



Université
de Toulouse

THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par :

Institut National Polytechnique de Toulouse (INP Toulouse)

Discipline ou spécialité :

Image, Information et Hypermédia

Présentée et soutenue par :

M. QUOC VIET DANG

le lundi 22 septembre 2014

Titre :

SIMILARITES DANS DES MODELES BRep PARAMETRIQUES:
DETECTION ET APPLICATIONS

Ecole doctorale :

Mathématiques, Informatique, Télécommunications de Toulouse (MITT)

Unité de recherche :

Institut de Recherche en Informatique de Toulouse (I.R.I.T.)

Directeur(s) de Thèse :

M. VINCENT CHARVILLAT

MME GÉRALDINE MORIN

Rapporteurs :

M. JEAN-CLAUDE LEON, INP GRENOBLE

M. MARC DANIEL, POLYTECH MARSEILLE

Membre(s) du jury :

M. MOHAMED DAOUDI, TELECOM LILLE, Président

M. BASILE SAUVAGE, UNIVERSITE DE STRASBOURG, Membre

Mme GÉRALDINE MORIN, INP TOULOUSE, Membre

Mme SANDRINE MOUYSET, UNIVERSITE TOULOUSE 3, Membre

M. VINCENT CHARVILLAT, INP TOULOUSE, Membre

*A mon père, ma mère et ma femme
Pour le soutien, l'amour et la confiance.*

Remerciements

Au premier lieu, j'adresse mes remerciements sincères à tous les membres du jury pour avoir consacré beaucoup de temps à l'évaluation de ce travail. Je tiens à remercier particulièrement Messieurs les professeurs Marc DANIEL et Jean-Claude LÉON pour avoir accepté d'être membres rapporteurs de thèse ; leur lecture très attentive et intéressée et leurs remarques judicieuses m'ont permis de considérer d'autres aspects de ma recherche. Merci à Monsieur le professeur Mohamed DAOUDI et Monsieur le maître de conférences Basile SAUVAGE pour avoir accepté d'être membres du jury.

J'exprime ma profonde reconnaissance à Monsieur le professeur Vincent CHARVILLAT, directeur de thèse, pour m'avoir accueilli dans son équipe. Mes remerciements les plus sincères vont également à Madame le maître de conférences habilité Géraldine MORIN pour son encadrement très suivi. J'apprécie ses conseils avisés, sa pédagogie, ses encouragements, ses soutiens permanents permettant de mener cette thèse à son terme. Je tiens à remercier Madame le maître de conférences Sandrine MOUYSET pour ses conseils avisés, sa gentillesse et sa patience manifestées à mon égard durant cette thèse.

Je tiens à remercier le Ministère de l'Éducation et de la Formation du Vietnam et l'Institut National Polytechnique de Toulouse pour avoir financé ma thèse de doctorat. Mes remerciements vont également à l'Université de Can Tho pour son soutien administratif.

J'adresse mes salutations cordiales à tous les membres et amis de l'équipe VORTEX à l'INP-ENSEEIHRT pour l'accueil chaleureux pendant mon séjour, en particulier, Pierre GURDJOS et Sylvie CHAMBON pour les conseils avisés.

Je n'oublie pas de remercier mes amis vietnamiens (Thây Thai, Hoà, Nhi, Vân, Thanh Thanh, Huân, Lê, Dr Thao, Ly et les autres qui m'excuseront de ne pas les nommer) qui m'ont apporté l'amitié et des souvenirs inoubliables durant toutes ces années en France. Grâce à vous, j'ai retrouvé l'ambiance familiale à un endroit très loin de mon pays natal.

Enfin, je porte une attention affectueuse toute particulière à ma famille. D'abord à mes parents pour leurs soutiens moraux et financiers durant mes études. Ils n'ont point cessé de m'encourager de poursuivre mes études supérieures surtout dans le domaine de la recherche scientifique. Qu'ils veuillent trouver ma gratitude et mes reconnaissances. J'adresse également un grand merci à ma soeur, My Hanh, pour le soin de la famille au Vietnam pendant mon absence.

Mes derniers remerciements s'adressent à ma femme, Anh Thu, qui a assisté quotidiennement à l'avancement de cette thèse et qui m'a soutenu tous les jours, dans mes longues journées de travail, surtout dans les moments stressants de cette thèse. Qu'elle trouve mon amour et affection.

Résumé

Dans cette thèse, nous identifions et exploitons des similarités partielles dans des objets 3D pour répondre à des besoins courants du domaine de la Conception Assistée par Ordinateur (CAO). De nouvelles méthodes sont introduites, d'une part pour détecter les similarités partielles, d'autre part pour utiliser ces similarités dans des applications spécifiques telles que l'édition de forme, la compression et l'indexation d'objets 3D. Grâce au développement des applications de la modélisation géométrique, ces modèles sont de plus en plus nombreux et sont disponibles à travers plusieurs modalités. Pour augmenter la productivité dans la création de tels objets virtuels, la réutilisation et l'adaptation des modèles existants est un choix prioritaire. Cela exige donc des méthodes facilitant le stockage, la recherche et l'exploitation de ces modèles. Heureusement, les similarités dans des objets 3D est un phénomène fréquent. De nombreux objets sont composés de parties similaires à une rotation, à une translation ou à une symétrie près. De ce fait, la détection des similarités partielles dans ces modèles est capable de répondre aux problématiques courantes : la taille du stockage est réduite en conservant seulement une partie au lieu de toutes les parties répétées d'un modèle ; l'indexation des modèles 3D requiert a priori l'orientation canonique des modèles. Or, la symétrie dans un objet 3D est toujours une référence d'orientation cohérente avec la perception humaine. Nous utilisons donc la symétrie partielle pour aligner ces modèles et ainsi renforcer la robustesse des méthodes d'indexation.

Dans un premier temps, nous introduisons une approche similaire à la Transformée de Hough pour détecter des similarités partielles dans des modèles BRep-NURBS. Cette approche identifie non seulement les parties similaires mais aussi les transformations qui les lient. À travers la classification des isométries dans l'analyse des transformations, notre approche peut distinguer la nature de transformation liant des parties similaires d'un modèle, c'est-à-dire, les parties similaires à une rotation, à une translation ou à une symétrie près. Dans le deuxième temps, nous proposons deux applications héritées directement des résultats obtenus par la détection. Tout d'abord, pour la compression, un modèle se transforme en un graphe de similarités d'où les faces principales à conserver sont sélectionnées dans la structure compressée. Ensuite, pour l'orientation, le plan de la symétrie dominante et la projection orthographique d'un modèle autour de ce plan permettent de définir un repère canonique pour aligner ce modèle.

Mots clés : Modélisation géométrique, NURBS, représentation par bords, similarité, alignement 3D, compression 3D, édition 3D, indexation 3D.

Similarities within BRep Parametric Models : Detection and Applications.

Abstract : In this thesis, we identify and exploit the partial similarities within 3D objects to answer the current needs of the Computer Aided Design field (CAD). Novel methods are introduced, on the one hand to detect the partial similarities, on the other hand to use these similarities for specific applications such as shape editing, compression and indexation of 3D objects. Because of the development of geometric modeling applications, 3D models are getting more numerous and available through many channels. To increase the productivity in creating such 3D virtual objects, the reuse and the adaptation of existing models becomes a prior choice. Thus, it requires methods easing the storage, the searching and the exploitation of these models. Fortunately, similarities within the 3D objects is a popular phenomenon. Many objects are composed of similar patches up to an approximated rotation, translation or symmetry. Hence, detecting the partial similarities within NURBS-BRep models is able to solve the current issues : the storage size is reduced by coding a single patch instead of repeated patches of a model ; 3D model indexation requires a canonical orientation of these models. Furthermore, the symmetry within a 3D object is a good orientation reference, coherent with the human perception. Accordingly, we use the partial symmetries to align 3D models and so reinforce the robustness of indexation methods.

In a first phase, we introduce an original approach similar to the Hough Transform to detect partial similarities within NURBS-BRep models. This approach identifies not only similar patches but also identifies the corresponding transformations that connect them. Additionally, through the classification of isometries in transformations analysis, our approach can distinguish the nature of transformations of similar patches of a model, that is, the patches similar up to an approximated rotation, translation or symmetry. This classification is advantageous for further applications : the similar patches of other transformation natures are considered in compressing ; the symmetric patches are used to normalize 3D models aim at a robust indexation. In the second phase, we propose two applications inherited directly from the obtained results of the detection. Firstly, for the compression, a model is transformed into a similarity graph where the principal faces to be coded are selected to form the compressed structure. Secondly, for the orientation, the plane of the dominant symmetry and the orthographic projection of a model around this plane generate a canonical frame to align this model.

Keywords : Geometric modeling, NURBS, boundary representation, similarity, 3D alignment, 3D compression, 3D edition, 3D indexation.

Table des figures

1.1	Un extrait du court-métrage d'animation <i>Geri's game</i> et la modélisation du personnage par surface de subdivision. ¹	2
1.2	Capture d'écran du modelleur Rhinocéros dans la conception 3D d'un appareil photo par les surfaces NURBS.	2
1.3	Différentes utilisations des logiciels MG.	3
1.4	Visualisation du cerveau par X Toolkit.	4
1.5	Les similarités existantes dans les modèles 3D.	5
2.1	Supports numériques permettant d'acquérir les cartes de profondeurs.	10
2.2	Reconstruction 3D d'un cerveau.	11
2.3	Captures d'écran d'un logiciel CAO et d'un modelleur 3D.	11
2.4	Un modèle polygonal.	12
2.5	Exemples de modélisation surfacique.	13
2.6	Modélisation volumétrique.	14
2.7	Courbe et surface NURBS.	16
2.8	Structure de données d'un objet BRep.	18
2.9	Représentation par Bords dans le contexte d'OpenNURBS.	20
2.10	Exemple d'un modèle BRep d'un avion.	21
2.11	Échantillonnage équi-paramétrique d'une face.	22
2.12	Résultat de l'échantillonnage d'une face.	25
2.13	Résultats de l'échantillonnage de distance relative uniforme.	26
2.14	Illustration du calcul de coordonnée barycentrique.	27
2.15	Les boîtes englobantes de deux faces.	28
3.1	Symétrie planaire intrinsèque et extrinsèque.	44
3.2	Réflexion d'un modèle de champignon et son histogramme d'orientation correspondante [Sun 1997].	45
3.3	Calcul de la distance de symétrie [Zabrodsky 1995].	46
3.4	Les descripteurs de réflexion de quelques modèles [Kazhdan 2003].	46
3.5	Descripteur SIFT pour détecter la symétrie dans une image [Loy 2006].	47
3.6	Niveau de la réflexion capturé par PRST.	48
3.7	La détection de similarités dans le travail de Mitra [Mitra 2006].	48
3.8	La traduction des modèles BRep en modèles de point discrets.	49
3.9	Construction du plan de symétrie.	50
3.10	Les limitations dans l'algorithme proposé dans [Tate 2003].	50
3.11	Détection de symétrie dans des modèles BRep [Li 2011].	51
3.12	Procédé par vote.	52
3.13	Décomposition d'un modèle BRep-NURBS.	53
3.14	Algorithme proposé.	54
3.15	Extraction d'isométries basée topologie.	56

3.16	Extraction des isométries à partir d'une grille de points de la surface.	60
3.17	Transformation entre deux points p^i et p^j .	62
3.18	Orientation des tangentes principales.	63
3.19	Exemple sur l'évaluation des caractéristiques de transformations.	64
3.20	Incohérence entre deux repères locaux symétriques.	64
3.21	Réorientation par le voisinage.	64
3.22	Principe de la méthode Mean Shift [Comaniciu 1999].	68
3.23	Illustration des étapes de la classification spectrale.	70
3.24	Principe de l'heuristique (3.27).	71
3.25	Principe de l'heuristique pour le choix du nombre de classes \hat{k} (3.29).	72
3.26	Résultats de Mean Shift et Spectrale.	73
3.27	Le voisinage d'un point dans la grille d'échantillons.	75
3.28	Isométries indirectes détectées.	78
3.29	Résultats de la classification.	78
3.30	Résultat de l'extraction basée géométrie.	79
3.31	Résultats de la détection dans des composantes de CAO. ²	79
3.32	Résultats de la détection dans des modèles complexes.	80
3.33	Cas dégénérés.	81
4.1	Alignement basé sur l'ACP.	84
4.2	Alignement avec NPCA et CPCA.	85
4.3	Résultats de l'alignement basé symétrie.	86
4.4	Six vues canoniques de la projection orthographique.	86
4.5	Différentes vues de la projection orthographique de deux objets 3D.	86
4.6	Résultats de l'alignement par MPA.	87
4.7	Algorithme de détection de symétrie partielle dominante.	88
4.8	Les symétries partielles dominantes détectées.	89
4.9	Aires de projection du modèle d'une voiture.	89
4.10	Résultats de l'alignement des modèles BRep-NURBS.	90
4.11	Extraction des silhouettes d'un modèle.	92
4.12	Résultats de trois requêtes.	93
4.13	Résultats de trois requêtes.	94
4.14	L'algorithme de compression de Furukawa et al. [Furukawa 2002].	96
4.15	Modèles à tester la compression.	97

Liste des tableaux

2.1	Équivalence entre les spécifications d'OpenNURBS et de BRep Théorique.	21
2.2	Groupes d'isométries dans l'espace vectoriel de dimension 3.	32
2.3	Nature des points fixes, en supposant que f a au moins un point fixe.	34
2.4	Liste des classes d'isométries.	37
2.5	Illustrations des classes d'isométries.	38
2.6	Invariants des classes d'isométries.	39
3.1	Temps d'exécution de chaque étape de l'algorithme.	80
4.1	Résultats de la compression.	97

Liste des Abréviations

ACP	Analyse en Composantes Principales
AMP	Alignement par Minimisation de Projection
BRep	Boundary Representation
BRep-NURBS	BRep basé NURBS
CAO	Conception Assistée par Ordinateurs
CPCA	Continuous Principal Component Analysis
CSG	Constructive Solid Geometry
CT-scan	Computed Tomography scan
IRM	Imagerie par Résonance Magnétique
MG	Modélisation Géométrique
NPCA	Normalized Principal Component Analysis
NURBS	Non-Uniform Rational B-Spline
PET-scan	Positron Emission Tomography scan
PRST	Planar-Reflective Symmetry Transform
SIFT	Scale-Invariant Feature Transform
SVD	Singular Value Decomposition

Table des matières

1	Introduction	1
1	Modélisation Géométrique : contexte global	1
2	Contexte de la recherche	4
3	Approches proposées	5
4	Vue générale	7
2	Isométries et Modèles BRep	9
1	Modélisation des objets 3D	9
1.1	Production des données 3D	9
1.1.1	Acquisition	10
1.1.2	Création manuelle	11
1.2	Modélisation polygonale	12
1.3	Modélisation surfacique	12
1.4	Modélisation volumique	13
2	Modèle BRep basé NURBS	14
2.1	Courbe et surface NURBS	15
2.1.1	Définition	15
2.1.2	Pourquoi utiliser les NURBS ?	16
2.2	Représentation par Bords	17
2.3	BRep dans le contexte d'OpenNURBS	20
2.4	Échantillonnage des Faces	22
2.4.1	Échantillonnage à l'intérieur de la face	23
2.4.2	Échantillonnage de distance relativement uniforme	25
2.4.3	Échantillonnage par pondération	26
2.4.4	Conclusion	27
3	Isométrie : caractérisation de similarité	28
3.1	Isométries dans l'espace tridimensionnel	29
3.1.1	Isométries linéaires dans l'espace vectoriel Euclidien	30
3.1.2	Isométries affines dans l'espace Affine Euclidien	33
3.2	Groupe des isométries partielles	38
3.2.1	Classification des isométries	39
3.2.2	Métriques des isométries	39
4	Conclusion	42
3	Détection d'isométries partielles	43
1	Travaux précédents	44
1.1	Procédé de type transformée de Hough	46
1.2	Détection de similarités dans des modèles BRep	49
1.3	Conclusion	51
2	Algorithme de détection	52

2.1	Utilisation de la transformée de Hough	52
2.2	Algorithme proposé	54
3	Extraction des isométries	54
3.1	Extraction basée topologie	55
3.1.1	Définitions	55
3.1.2	Méthode générale	56
3.1.3	Mise en correspondance des faces	57
3.1.4	Estimation des isométries	58
3.2	Extraction basée géométrie	60
3.2.1	Signatures des points de la surface paramétrique	60
3.2.2	Orientation dans la quadrille paramétrique	62
3.2.3	Analyse du calcul direct des transformations	62
3.3	Caractérisation des isométries	65
3.4	Conclusion	66
4	Classification des isométries	66
4.1	Méthode Mean Shift	66
4.2	Méthode Spectrale	69
4.3	Application de la classification spectrale pour la recherche de similarités	73
5	Validation	73
5.1	Validation itérative	74
5.2	Validation par pondération	74
5.3	Validation basée histogramme	75
5.4	Validation basée distance de Hausdorff	76
6	Extension	77
7	Expérimentations	77
8	Conclusion	81
4	Application de la détection d'isométries partielles	83
1	Alignement des modèles 3D	83
1.1	Introduction	83
1.2	Approche	87
1.2.1	Détection de la réflexion partielle dominante	88
1.2.2	Rectification du repère de référence	89
1.2.3	Expérimentations	91
1.3	Application à l'indexation des modèles BRep-NURBS	91
1.4	Conclusion	94
2	Compression des modèles	94
2.1	Introduction	94
2.2	Travaux précédents	95
2.3	Approche proposée	96
2.4	Conclusion	98

5	Conclusion et Perspectives	99
1	Conclusion	99
2	Perspectives	101
2.1	Détection de similarités	101
2.2	Applications basées sur les similarités détectées	101
	Bibliographie	103

Introduction

Sommaire

1	Modélisation Géométrique : contexte global	1
2	Contexte de la recherche	4
3	Approches proposées	5
4	Vue générale	7

1 Modélisation Géométrique : contexte global

Le domaine de la *Modélisation Géométrique* (MG) a commencé à se développer vers les années 1960 dans l'industrie automobile, au moment où les ordinateurs ont proposé des outils permettant de générer des modèles numériques pour la production : de la conception à la fabrication. Dans [Farin 1992], Farin résume une histoire très détaillée de la MG depuis l'ère Romaine. Il explique aussi que les polynômes sont choisis comme un modèle numérique pour la représentation des courbes et des surfaces de *formes gauches*, tout d'abord par de Casteljau, ensuite et indépendamment par Bézier. Les fonctions polynômes sont exprimées par des polynômes de Bernstein pour définir les courbes et les surfaces de Bézier en fonction des points de contrôle (en tant que coefficients des polynômes de Bernstein). Puis, les *polynômes par morceaux* sont considérés pour modéliser les formes plus complexes sans augmenter le degré et ils permettent aussi la modification locale par les points de contrôle. Leurs représentations canoniques sous forme B-Spline étaient utilisées par Schoenberg [Schoenberg 1968] pour lisser les données statistiques. Alors, la communauté de la MG développe différents modèles de courbes et de surfaces, et fournit des outils associés pour générer des objets de *forme gauche*. Les logiciels et les techniques de la MG se regroupent sous le nom de *Conception Assistée par Ordinateur* (CAO), un domaine dont les évolutions ont été transférées dans l'industrie. Les *surfaces de subdivision* sont aussi utilisées dans l'industrie des films d'animation : le film *Geri's game* de Pixar en 1997 [DeRose 1998] est un exemple de succès des *modeleurs* basés sur des surfaces de subdivision (figure 1.1) ; les *NURBS* (B-Spline Non-Uniforme Rationnelle) sont acceptées comme un standard pour la conception de modèles paramétriques dans l'industrie (figure 1.2).

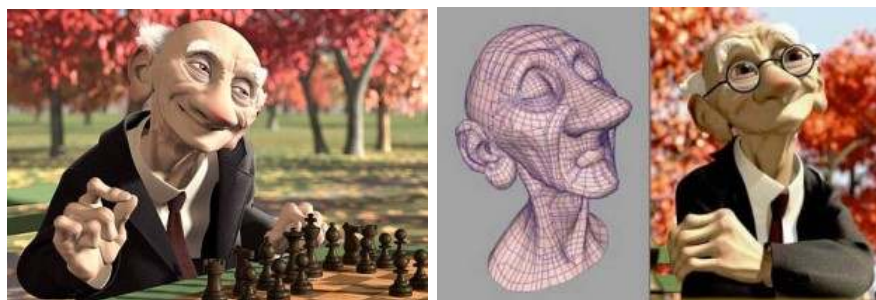


FIGURE 1.1 – Un extrait du court-métrage d'animation *Geri's game* et la modélisation du personnage par surface de subdivision.¹

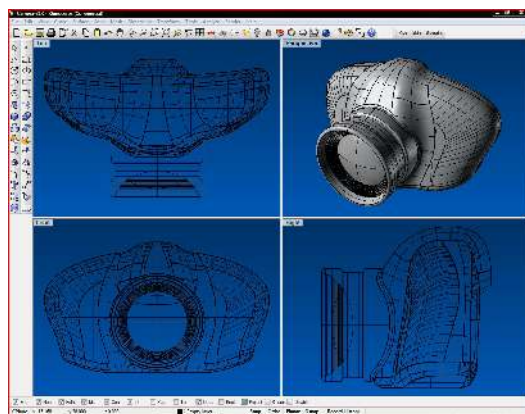


FIGURE 1.2 – Capture d'écran du modelleur Rhinocéros dans la conception 3D d'un appareil photo par les surfaces NURBS.

De nos jours, après de nombreux développements et grâce aux émergences technologiques et aux avancées de la recherche dans ce domaine, on a recours de plus en plus à la MG. En effet, de nombreux outils performants ont été développés et commercialisés, dédiés à des applications industrielles comme la conception et la production de produits, la visualisation, l'analyse des surfaces et la réalité virtuelle. Ces applications sont utilisées dans plusieurs domaines telles que l'électricité, la mécanique, la biologie, le cinéma [Dimas 1999]. Avec la capacité d'abstraction des objets physiques en objets numériques, la MG rend la production industrielle moins coûteuse et plus efficace. C'est pour cette raison que nous la retrouvons partout dans l'industrie.

Les logiciels de la MG évoluent du contexte professionnel aux applications Grand Public. Les fournisseurs classiques des logiciels CAO développent de nouveaux produits. Par exemple, Daussault introduit *Imagine and Shape* appliquant les surfaces de subdivision pour la conception des prototypes (figure 1.3a). Le but est de développer une interface intuitive dissimulant la théorie mathématique sous-jacente aux utilisateurs. Pourtant, les utilisateurs doivent être professionnels dans le domaine

1. Images de courtoisie de Hannah <http://www.pixartalk.com/shorts/geris-game/>

ou être formés pour l'utiliser avec efficacité. Dans la médecine, les techniques de visualisation évoluent rapidement grâce aux modèles 3D et aux outils de visualisation associés, par exemple Geneva (figure 1.3b).



FIGURE 1.3 – Différentes utilisations professionnelles des logiciels en Modélisation Géométrique. (a) Conception d'un aspirateur au sein de *Imagine and Shape*. (b) Chirurgie assistée par la visualisation 3D dans les Hôpitaux Universitaires de Genève (Geneva).

Comme les technologies progressent considérablement, les modèles 3D ne restent plus seulement à la portée des utilisateurs professionnels mais ils sont aussi accessibles aux amateurs. Paru en 2003, *Second life* permet aux utilisateurs de créer et de partager les contenus 3D. En 2009, *Khronos group* a proposé le standard gratuit *WebGL* pour insérer les primitives 3D aux navigateurs web. WebGL contribuera sans doute au développement des applications dans plusieurs domaines. Appliquée au e-Commerce, cette technologie pourrait permettre d'offrir des vues 3D des produits avec un maximum de détails : faire tourner et pivoter un objet à 360 degrés, avoir la possibilité de zoomer de manière très précise. Appliquée aux autres domaines scientifiques, cette technologie supporte la visualisation scientifique dans des navigateurs web comparables aux logiciels dédiés : par exemple, Daniel Haehn a créé *X Toolkit* pour la visualisation et l'interaction avec des images médicales² (figure 1.4).

2. <http://www.goXTK.com>

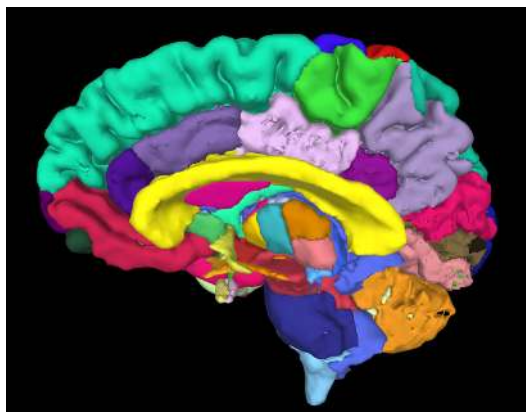


FIGURE 1.4 – Visualisation du cerveau par X Toolkit.

Malgré son développement énorme ces dernières décennies, la MG reste toujours un domaine de recherche actif et propose aussi de nouveaux défis pour s'intégrer dans la vie quotidienne et pour développer des applications visant une communauté d'utilisateurs plus large.

2 Contexte de la recherche

Dans le domaine de la Conception Assistée par Ordinateur (CAO), il existe un besoin de très haute précision dans la création des modèles 3D [Dugelay 2008]. Les courbes et les surfaces paramétriques sont utilisées en priorité. En particulier, les NURBS fournissent des outils puissants aux communautés académiques et industrielles pour la conception et l'analyse des objets en 3D [Dimas 1999]. Initiées par des recherches industrielles dans les années 1980 chez Boeing, SDRC et l'Université d'Utah [Piegl 1991], les courbes et les surfaces NURBS sont de plus en plus utilisées dans le domaine de CAO. Depuis, les intérêts dans l'évaluation et la manipulation apportés par leurs représentations paramétriques définies par pôles, ont séduit le monde industriel. Avec un grand nombre de recherches scientifiques, les NURBS sont entrées dans les standards. De nombreux logiciels basés NURBS dédiés à la conception 3D industrielle tels que AutoCAD, CATIA, Solidworks, Rhinoceros sont développés et proposés dans plusieurs domaines : architecture, aérospatiale, ingénierie et construction, médical et produits de consommation. De nos jours, les NURBS restent toujours un sujet actuel de recherche pour répondre aux nouvelles exigences.

Les modèles dans la CAO sont modélisés suivant deux méthodes [El-Mehalawi 2003] : la modélisation volumique et la modélisation surfacique. Les deux représentants significatifs de ces méthodes sont respectivement **CSG** (Constructive Solid Geometry en anglais) et **BRep** (Boundary Representation en anglais). Alors que les BRep décrivent les modèles en fonction des sommets, des arêtes et des surfaces, la CSG les décrit en fonction des opérations booléennes (union, intersection, soustraction) appliquées sur des primitives (cylindre, cube, sphère, etc.). Grâce aux opérateurs booléens et aux primitives, la méthode CSG semble la plus aisée à utiliser dans

les applications CAO. Par contre, elle est moins préférable et moins utilisée que la méthode B-Rep [Stroud 2011] dont les opérations sont plus riches et les formes des objets plus variées. Alors, associés aux surfaces NURBS, les modèles BRep basés NURBS deviennent un standard et donnent des outils CAO de haute performance et ont contribué à plusieurs succès dans de nombreux domaines tels que la chimie moléculaire [Bajaj 1997], les systèmes d'information géographique [Caumon 2003] et en particulier la conception des composantes mécaniques [Chu 2006, Cuillière 2011].

Comme le nombre de modèles 3D évolue au cours du temps, les répertoires de ces modèles progressent sans cesse. De plus, les concepteurs 3D ont tendance à adapter les modèles existants pour adresser les nouveaux besoins. En effet, d'après une statistique [Iyer 2005], plus de 75% des activités de conception consiste à la réutilisation ou à l'adaptation des conceptions existantes. De ce fait, il est nécessaire de concevoir des méthodes et mécanismes pour avoir un stockage optimisé et des recherches efficaces. Dans ce contexte, en analysant et exploitant les propriétés géométriques et topologiques des modèles BRep basés NURBS, nous souhaitons introduire et générer des caractéristiques qui peuvent servir à la compression et à l'indexation des modèles dans de grands répertoires.

3 Approches proposées

La **similarité** est un phénomène fréquent existant dans des objets naturels et synthétiques. De nombreux objets possèdent des parties similaires à une rotation, une translation, une réflexion ou plus généralement à une isométrie près. La figure 1.5 illustre les similarités dans deux modèles 3D : le modèle du dinosaure expose la réflexion entre deux parties du corps et le modèle du château Chambord expose la translation ou la réflexion entre les tours. Dans le domaine de la MG, plusieurs travaux de recherche concernant la similarité ont été réalisés sous la notion de **symétrie**. En terme de vocabulaire, cette notion peut entraîner des confusions avec la **réflexion** en mathématique géométrique. Pourtant, la **symétrie** n'est pas tout à fait une réflexion !

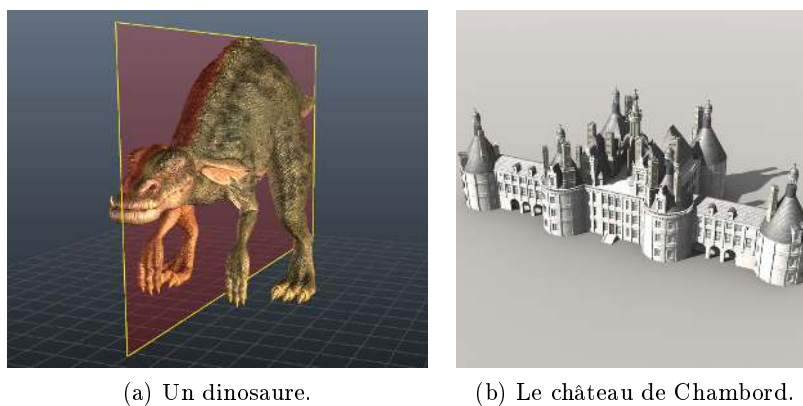


FIGURE 1.5 – Les similarités existantes dans les modèles 3D.

La **symétrie** est un concept commun utilisé largement dans la vie quotidienne. Nous pouvons la trouver partout et à toute échelle [Hel-Or 2010, Mitra 2012] : des objets naturels aux objets artificiels, des nanostructures, des organisations biologiques et aussi des œuvres artistiques. L'être humain possède une excellente capacité d'identification et de reconnaissance de symétries afin de les utiliser dans divers domaines. En effet, Hermann Weyl a confirmé dans son ouvrage [Weyl 1952] : "*la symétrie est une parmi des idées avec lesquelles l'être humain de toute âge essaie d'apercevoir et de créer l'ordre, la beauté et la perfection*". De ce fait, nous constatons que la symétrie est une caractéristique essentielle de la perception humaine. De plus, les recherches en informatique visent à imiter et à intégrer les capacités humaines dans des techniques existantes pour bien résoudre des problèmes, simples ou compliqués, avec une haute précision. Alors, la recherche de symétries en informatique peut-elle donner des résultats réalistes et utiles, satisfaisant nos besoins actuels ? Pourtant, mathématiquement, qu'est-ce qu'une symétrie ? Quels sont les problèmes posés par la recherche des symétries ? Quelles sont les pistes possibles de l'application des symétries ? Nous allons tenter de répondre à ces questions dans ce travail de recherche.

D'une façon générale, la notion de *symétrie* concerne les choses qui sont bien équilibrées, bien proportionnées. Classiquement, un objet est dit symétrique s'il admet une réflexion propre entre deux parties de cet objet séparées par un miroir. Donc, il y a une ligne divisant deux parties bien équilibrées pour les objets en 2D ou un plan pour les objets en 3D. Néanmoins, la géométrie moderne donne une interprétation plus générale : la conception de symétrie est algébriquement formalisée sous forme de *transformations*. La symétrie ne reste plus une simple réflexion mais permet de nouvelles notions [Weyl 1952, Cederberg 2001] : la *symétrie bilatérale*, la *symétrie rotationnelle*, la *symétrie translationnelle*, la *symétrie cristallographique*, etc. La définition suivante formalise cette notion :

Définition 1 Une transformation S est dite **symétrie** d'un ensemble de points α si ces points sont invariants sous l'application de S , autrement dit, $S(\alpha) = \alpha$. Si S n'est pas une transformation identité ($S \neq I$), α sera symétrique ou bien α aura une symétrie S .

Cette définition mentionne que la symétrie est une transformation préservant la structure d'un ensemble de points [Cederberg 2001]. Ainsi, cette définition exige de définir les moyens d'analyser les invariants d'une transformation et la structure des points dans un espace a priori.

Remarque 1 Nous n'exploitons pas les notions des groupes de symétries. Ainsi, nous voulons éviter les confusions en introduisant nos approches proposées pour résoudre les problèmes dans le contexte de notre travail. À partir de ce paragraphe, nous utilisons la notion de **similarité** au lieu de symétrie. La symétrie est donc équivalente à la réflexion.

Dans cette thèse, nous proposons d'exploiter les similarités existantes dans des modèles BRep basés NURBS (BRep-NURBS) en vue de deux applications :

- L'**alignement** des modèles BRep-NURBS en utilisant la symétrie partielle dominante. Cette étape de pré-traitement est très importante pour rendre les méthodes d'indexation des modèles 3D plus efficaces.
- La **compression** des modèles BRep-NURBS, au lieu de stocker tout un modèle, seulement une des parties similaires est stockée.

En vue de ces applications, notre travail présente les contributions suivantes :

- Un algorithme du style de Hough pour la détection de similarités en utilisant la structure topologique des modèles BRep-NURBS.
- L'application des points fixes pour identifier les classes d'isométries de tous types de similarités.
- Une approche originale en combinant la symétrie dominante et l'aire de projection minimale pour l'alignement.

4 Vue générale

Le reste de cette thèse est organisé comme suit :

- Le chapitre 2 présente la théorie fondamentale de cette thèse. Les différents moyens de production des données 3D entraînent différentes méthodes de modélisation des objets 3D. Ici, les trois méthodes connues dans la MG seront présentées : la modélisation polygonale, la modélisation surfacique et la modélisation volumique. À travers les analyses de ces méthodes, nous expliquerons notre choix de modélisation des objets 3D utilisé dans cette thèse : la Représentation par bords basées NURBS (BRep-NURBS). Ces modèles 3D seront décrits en détails dans le contexte d'OpenNURBS, la librairie que nous utilisons dans les expérimentations. Ainsi, nos méthodes d'échantillonnage dans les faces de l'objet BRep-NURBS seront introduites, ces méthodes sont nécessaires pour notre algorithme de détection des similarités partielles. Ensuite, les notions et les analyses des isométries dans l'espace Affine Euclidien, le facteur principal qui conduit au succès de ce travail de recherche, seront présentées : la classification, la caractérisation et la métrique.
- Le chapitre 3 présente la contribution principale de cette thèse : la détection des isométries partielles dans des modèles BRep-NURBS. Les travaux précédents seront tout d'abord résumés. Nous adaptons le *Procédé de la transformée de Hough* dans notre algorithme. L'idée principale de cet algorithme est l'utilisation des *loops* extérieures pour identifier les régions similaires et pour estimer les isométries correspondantes. Comme les topologies des faces sont très variées, il existe des modèles ayant une seule face. Nous proposons une approche supplémentaire pour résoudre ce problème en privilégiant le domaine des points discrets. Alors, notre algorithme admet deux approches d'extraction des isométries : basée topologie et basée géométrie. Puis, nous présenterons notre approche de classification non supervisée utilisée dans l'algorithme. Ainsi, en profitant de la structure des *loops* extérieures, nous proposons différentes méthodes pour la validation des isométries entre deux

faces similaires. Enfin, les expérimentations sur la détection des similarités seront présentées pour illustrer l'efficacité de notre algorithme.

- Le chapitre 4 présente une application des similarités détectées : l'alignement des modèles 3D. En effet, en utilisant la symétrie planaire partielle dominante d'un modèle BRep-NURBS, nous pouvons identifier le *repère de référence* pour l'aligner. L'algorithme proposé associe le plan de la symétrie partielle dominante et la projection orthographique du modèle. Ensuite, notre algorithme est appliqué à l'indexation des modèles BRep-NURBS comme une tâche de prétraitement. L'efficacité de notre algorithme sera évaluée via des courbes de précision/rappel. Ainsi, nous proposons une approche pour la compression des modèles, visant les *streaming des aperçus 3D*.
- Le manuscrit de cette thèse se finit par le chapitre 5 avec les conclusions et les perspectives de recherche sur la détection des isométries partielles et ses applications.

Isométries et Modèles BRep

Sommaire

1	Modélisation des objets 3D	9
1.1	Production des données 3D	9
1.2	Modélisation polygonale	12
1.3	Modélisation surfacique	12
1.4	Modélisation volumique	13
2	Modèle BRep basé NURBS	14
2.1	Courbe et surface NURBS	15
2.2	Représentation par Bords	17
2.3	BRep dans le contexte d'OpenNURBS	20
2.4	Échantillonnage des Faces	22
3	Isométrie : caractérisation de similarité	28
3.1	Isométries dans l'espace tridimensionnel	29
3.2	Groupe des isométries partielles	38
4	Conclusion	42

1 Modélisation des objets 3D

Suivant la méthode de production des données, un objet 3D est numérisé par un ensemble d'éléments de base comme des points, des voxels, des surfaces. Ainsi, les objets 3D doivent être modélisés conformément à ces éléments pour faciliter leur analyse et leur manipulation. Une fois modélisé par une méthode spécifique, un objet 3D est appelé *modèle 3D*. Généralement, il existe plusieurs méthodes de modélisation en fonction des données 3D et des objectifs d'utilisation. Dans ce paragraphe, nous ne souhaitons pas détailler toutes les méthodes de modélisation 3D, nous voulons présenter, de manière générale, différentes représentations connues et utilisées actuellement. Avant d'introduire ces modélisations d'objets 3D, les méthodes de production d'objets 3D sont introduites. Cette introduction consiste à expliquer pourquoi il existe plusieurs méthodes de modélisation 3D.

1.1 Production des données 3D

Les données 3D peuvent être obtenues depuis plusieurs modalités. D'après Lavoué (chapitre 1 du livre *3D Object Processing : Compression, Indexing and Watermarking* [Dugelay 2008]), il existe deux classes principales de méthodes pour générer

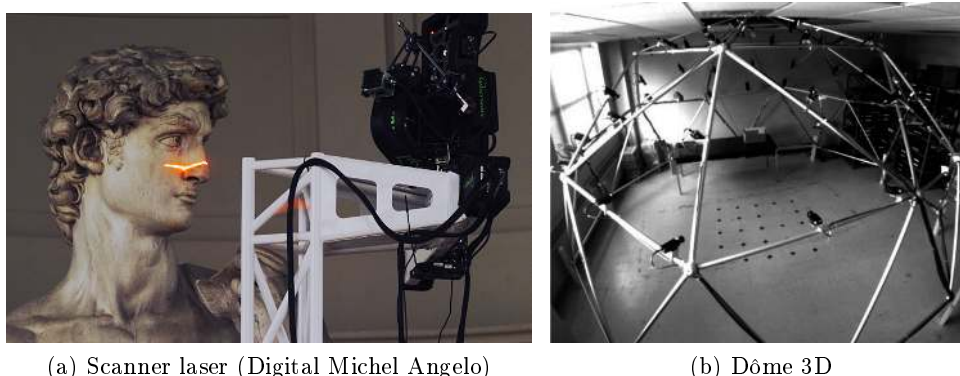


FIGURE 2.1 – Supports numériques permettant d'acquérir les cartes de profondeurs.

les données 3D : l'*acquisition* reproduit les données 3D à partir des objets réels et la *création manuelle* produit les données 3D en utilisant les logiciels spécifiques.

1.1.1 Acquisition

Grâce aux supports numériques, ces méthodes visent à reconstruire et numériser des objets réels dans le monde 3D. En se basant sur le type des supports, ces méthodes se divisent en deux groupes : l'un reconstruit la surface et l'autre reconstruit le volume de l'objet.

Le premier groupe utilise des machines capables de capturer l'objet réel, comme des appareils de photo sauf que des résultats ne sont plus des images en 2D mais des cartes de profondeurs estimées entre la machine et la surface de l'objet. Plusieurs captures consécutives autour de l'objet donnent un ensemble de cartes de profondeur à partir desquelles une tâche d'*alignement* des points de la surface est effectuée pour acquérir la forme complète en 3D. Par exemple, le projet *Digital Michelangelo* développé pendant l'année scolaire 1998-1999 à l'Université de Stanford [Koller 2004] utilise des scanners laser pour reconstruire les surfaces 3D de pièces de musées (figure 2.1a). Le *dôme 3D* développé à l'Université Carnegie Mellon en 1995 permet de capturer une séquence de poses d'un objet réel en mouvement, c'est-à-dire $3D + t$ (figure 2.1b).

Le deuxième groupe utilise les signaux pour récupérer les données depuis des objets réels : il est aussi appelé *la tomographie*. Les signaux les plus utilisés sont des rayons X dans le *CT-scan* (Computed Tomography scan), des champs magnétiques pour l'*IRM* (Imagerie par Résonance Magnétique) et l'émission radio active pour le *PET-scan* (Positron Emission Tomography scan). Pour numériser un objet, ces méthodes obtiennent tout d'abord des images 2D en le scannant successivement "tranche par tranche", un processus mathématique est ensuite lancé pour reconstruire les données 3D (figure 2.2). Contrairement au premier groupe, les données 3D obtenues de ces méthodes forment des grilles 3D volumiques.

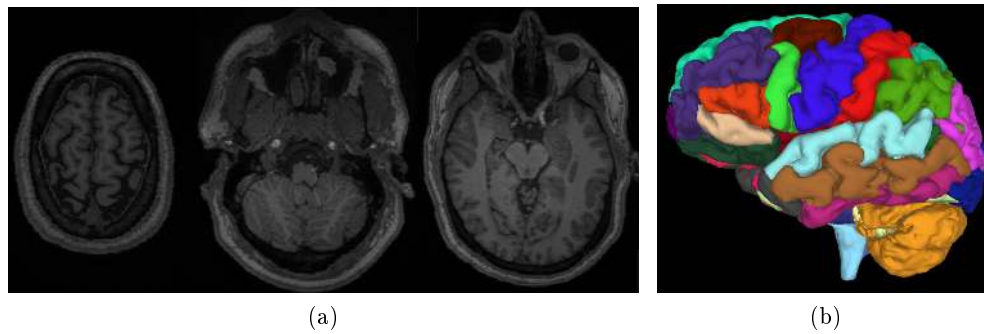


FIGURE 2.2 – Reconstruction 3D d'un cerveau (b) à partir des images obtenues par IRM (a).

1.1.2 Création manuelle

Les données 3D peuvent être aussi générées de façon synthétique par des utilisateurs de logiciels spécifiques. Il existe deux types de logiciels : le *logiciel CAO* et le *modèleur 3D*.

Les logiciels CAO sont principalement dédiés à l'ingénierie et à l'architecture. Ils supportent la création des objets 3D techniques avec une très haute précision en représentant mathématiquement ces objets. Ces logiciels sont utilisés largement dans la conception automobile, aéronautique, mécanique (figures 1.2, 2.3a). Les logiciels les plus connus sont TopSolid, CATIA, Rhinoceros, AutoCAD, SolidWorks.

Les modèleurs 3D permettent de produire des contenus 3D artistiques et des animations 3D. Ils supportent principalement la création des jeux et des vidéo 3D (figure 2.3b). Quelques logiciels connus sont Autodesk Maya, Blender, OpenFX.

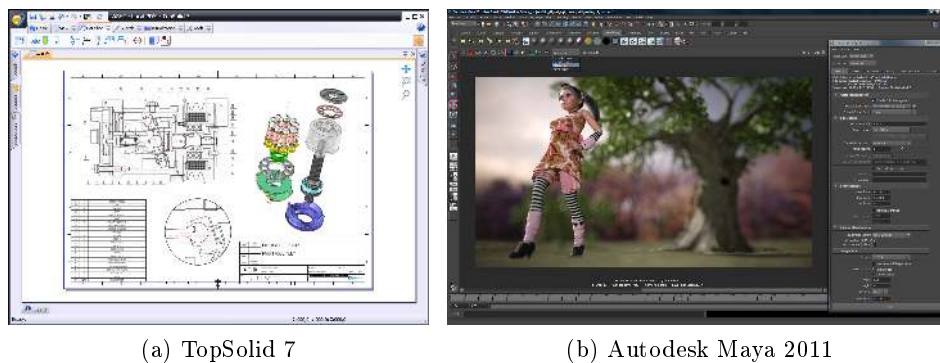


FIGURE 2.3 – Captures d'écran d'un logiciel CAO et d'un modèleur 3D.

À partir de là, nous constatons que les objets 3D peuvent être numérisés par plusieurs modalités. Les données 3D obtenues peuvent être un ensemble de points, une grille de volumes ou une représentation mathématique continue. Par conséquent, la MG requiert des méthodes de modélisation pour ces formes permettant l'analyse

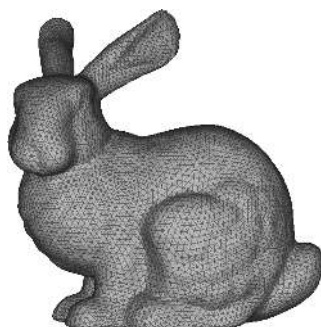


FIGURE 2.4 – Un modèle polygonal (maillage) ¹.

et la manipulation des objets 3D. Dans le paragraphe suivant, nous introduisons les modélisations connues dans le monde 3D qui sont divisées en trois groupes : la modélisation polygonale, la modélisation surfacique et la modélisation volumique.

1.2 Modélisation polygonale

Les modèles polygonaux sont des structures topologiques de faces (triangulaires ou quadrangulaires), aussi appelés des *maillages*. Ils se composent d'une partie géométrique et d'une partie topologique. Alors que la partie géométrique est un ensemble discret de sommets (points), la partie topologique consiste à définir les faces et les arêtes de la surface. Les maillages sont les modèles les plus utilisés car ils sont dérivés d'un ensemble d'échantillons de surfaces. Tous les maillages peuvent être facilement triangulés (chaque face a trois sommets). La triangulation est le seul modèle qui est nativement rendu par les cartes graphiques en utilisant les algorithmes implémentés sur les premiers matériels informatiques, rendant les maillages comme une représentation canonique des objets 3D.

1.3 Modélisation surfacique

Les *surfaces paramétriques* sont les plus utilisées dans les logiciels CAO [Farin 1992]. Ces surfaces sont des *produits tensoriels* soit de *polynômes par morceaux* soit de fonctions *polynomiales rationnelles par morceaux*. Elles sont aussi appelées surfaces de *forme gauche* et sont généralement lisses. Elles sont contrôlées par un ensemble de pôles appelés *points de contrôle*. Leurs continuités sont contrôlées par des vecteurs de noeuds et les degrés associés. Les surfaces de degré 3 ou 5 sont davantage utilisées. En général, les surfaces paramétriques sont employées dans les applications où il y a un besoin de haute précision et de bonne qualité au niveau des surfaces. Dans la conception industrielle, les surfaces NURBS sont devenues une représentation standard des surfaces paramétriques [Piegl 1991] (figure 2.5a) et sont utilisées dans plusieurs logiciels commercialisés tels que *CATIA* de Dassault System, *Rhinoceros* de Robert McNeel & Associates, *Creo* de Parametric Corporation Technologies. Une

1. Le lapin de Standford <https://graphics.stanford.edu/data/3Dscanrep/>.

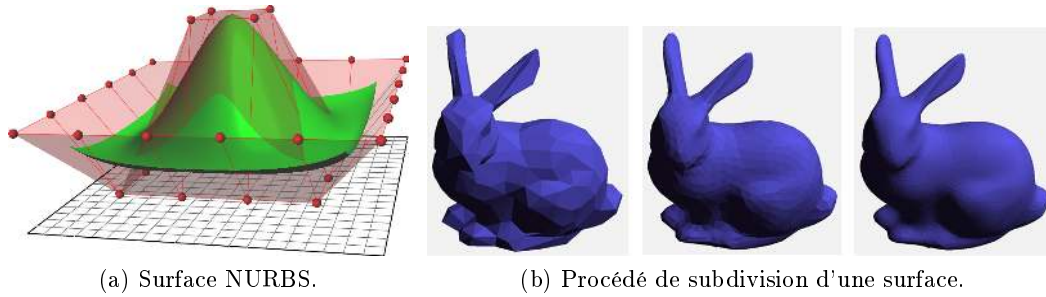


FIGURE 2.5 – Exemples de modélisation surfacique.

surface NURBS est composée de plusieurs morceaux qui sont l'image en 3D d'un domaine 2D paramétrique. Cette représentation paramétrique permet de générer facilement les points 3D de la surface en considérant des couples (u, v) dans le domaine paramétrique 2D. De plus, les informations différentielles sont naturellement disponibles et sont indépendantes du degré de la surface.

Un autre type de surface utilisé très souvent dans des applications de modélisation, en particulier pour la création de films d'animation, est la *surface de subdivision* [DeRose 1998] (figure 2.5b). Il y a un compromis dans la nature de ces surfaces entre *paramétrique* et *polygonale*. En effet, les surfaces de subdivision sont générées à partir de maillages de topologie quelconque avec le support des opérateurs de raffinement, c'est-à-dire, les maillages sont raffinés à une résolution arbitraire et convergent vers des surfaces lisses. Les algorithmes de subdivision les plus connus sont Catmull-Clark [Catmull 1978] et Loop [Loop 1987].

Certains objets 3D complexes ne peuvent pas être modélisés par une simple surface paramétrique. Comme la forme d'un objet dépend complètement de sa surface extérieure, la *Représentation par Bords* (BRep) est une technique utilisée comme un standard dans le domaine de la CAO. Pour modéliser un objet, cette représentation définit deux parties : les entités topologiques et les entités géométriques. Les entités géométriques sont des *surfaces*, *courbes* et *points*. Elles sont abstraites par des entités topologiques : *faces*, *arêtes* et *sommets*. Les sommets sont simplement des points sur une surface, une arête est une partie d'une courbe limitée par deux sommets et une face est une partie d'une surface limitée par plusieurs arêtes qui forment un circuit fermé. Il existe en plus deux autres entités topologiques : les *loops* et les *trims*. Dépendant du type de *loop* (intérieur ou extérieur), une *loop* détermine la portion limitée par des arêtes à être gardée ou être supprimée. Une *loop* est composée de plusieurs trims qui font correspondre les arêtes pour former des bords fermés. Cette représentation sera détaillée dans la section 2.

1.4 Modélisation volumique

Cette modélisation semble la plus naturelle pour représenter les objets 3D. En effet, le concept de *voxel* (volume élément) est connu depuis longtemps [Perlin 1985]. Un objet 3D est représenté par un ensemble de voxels donnant une approximation

de la forme de l'objet (image 2.6a). La modélisation volumique comprend aussi la représentation par particules adaptée à la simulation dans la mécanique des fluides (fumée, eau) et est particulièrement utilisée dans le domaine de la médecine car ces modèles sont directement reconstruits par la tomographie.

Un autre aspect de cette modélisation s'appelle la *modélisation des solides* dont la technique, très utilisée dans l'industrie, est la *CSG* (*Constructive Solid Geometry*, Géométrie de Construction de Solides en français). En définissant les opérateurs booléens (union, intersection, soustraction), cette technique représente un objet solide comme une combinaison des primitives telles que cylindre, cône, sphère, tore, cube. Un objet 3D est donc représenté par une structure arborescente où les primitives sont des feuilles et les opérateurs associent les noeuds intermédiaires (image 2.6b).

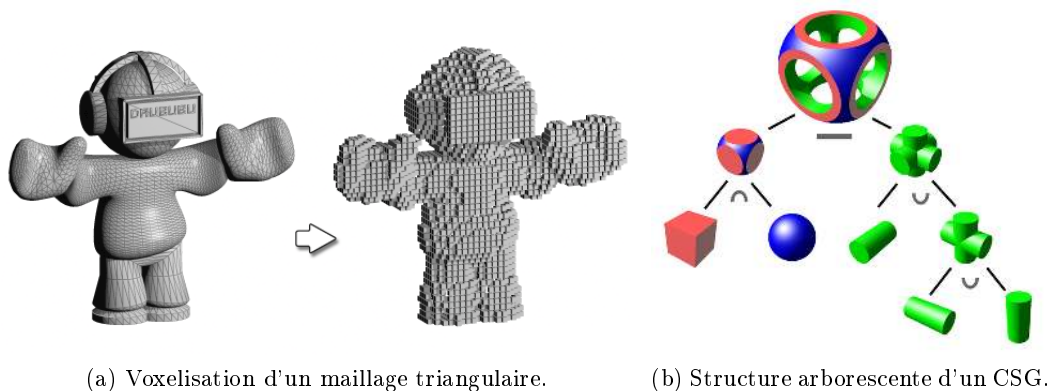


FIGURE 2.6 – Modélisation volumétrique.

Dans le contexte de notre recherche, nous n'avons pas l'ambition d'étudier toutes les méthodes de modélisation. Comme chaque méthode possède ses propres spécifications, il n'est pas possible de fusionner tous types de modèles dans un même processus de traitement. Dans ce travail, nous développons des applications pour répondre à quelques besoins actuels du domaine de la CAO. Nous nous intéressons donc aux modèles paramétriques, standards dans l'industrie. Dans le paragraphe suivant, nous détaillons le modèle choisi en analysant ses avantages et ses inconvénients.

2 Modèle BRep basé NURBS

Comme mentionné dans la section précédente, nous voulons développer des applications dédiées au domaine de la CAO. Ainsi, les modèles paramétriques sont appropriés aux demandes de haute précision lors de la conception et de la fabrication dans l'industrie. Considérées comme un standard industriel [Piegl 1991], les surfaces NURBS (Non-Uniform Rational B-Spline en anglais) sont notre point de départ. Cependant, les surfaces NURBS seules ne peuvent pas modéliser les objets 3D

de topologies arbitraires. Les modélisations surfaciques et volumiques nous offrent deux méthodes pour surmonter ce problème : la Représentation par Bords (BRep) et la Géométrie de Construction de Solides (CSG). D'après la section précédente, en comparaison aux CSG, les BRep présentent des avantages et nous choisissons les BRep en associant aux surfaces NURBS pour modéliser les objets 3D. En réalité, les modèles BRep basés NURBS (BRep-NURBS) sont utilisés dans plusieurs domaines tels que la *chimie moléculaire* [Bajaj 1997], les *systèmes d'information géographique* [Caumon 2003], la *conception des composantes mécaniques* [Chu 2006]. Dans la suite, nous introduisons des théories fondamentales comme les avantages de la surface NURBS et de la Représentations par Bords pour clarifier comment les modèles BRep-NURBS sont devenus un standard du domaine de la CAO.

2.1 Courbe et surface NURBS

Les courbes et surfaces NURBS constituent un outil classique pour les communautés académiques et industrielles pour la conception et l'analyse des objets 3D [Dimas 1999]. Dans [Piegl 1991], Piegl résume les caractéristiques importantes qui ont amené les NURBS à devenir un standard de la représentation et de la conception géométrique. Historiquement, les surfaces NURBS ont été développées et utilisées dans l'industrie vers les années 1970s. En se basant sur des courbes de Bézier, des courbes et des surfaces B-Splines étaient considérées comme des courbes et des surfaces de Bézier par morceaux. Elles étaient ensuite développées en B-Splines Rationnelles, introduisant des NURBS. Les premiers logiciels CAO dédiés à la conception des surfaces NURBS ont été : Tiger System de Boeing en 1979, Geomod de Structural Dynamics Research Corporation en 1983, Alpha-1 de l'Université d'Utah en 1984 et I/EMS d'Intergraph Corporation en 1984. Les succès de ces logiciels ont lancé les NURBS dans les standards comme IGES (International Graphics Exchange Standard) et PHIGS (Programmer's Hierarchical Interactive Graphics System).

2.1.1 Définition

Étant donnés un vecteur de noeuds $T = \{t_0, \dots, t_m\}$ et un ensemble de points de contrôle $P^w = \{P_0, \dots, P_n\}$ où chaque point $P_i = (x_i, y_i, z_i)$ est pondéré par un poids $w_i \in \mathbb{R}$ non nul. La courbe NURBS $C(t)$ de degré d est définie par l'équation [Piegl 1987] :

$$C(t) = \frac{\sum_{i=0}^n N_{i,d}(t)w_i P_i}{\sum_{i=0}^n N_{i,d}(t)w_i} \quad (0 \leq t \leq 1) \quad (2.1)$$

où $N_{i,d}$ sont des fonctions NURBS de degré d . Le degré, le nombre de noeuds et le nombre de points de contrôle sont liés à la formule : $m = n + d + 1$. Soit le vecteur de noeuds $T = \{t_0, \dots, t_i, t_{i+1}, \dots, t_m\}$, les fonctions NURBS de degré d (ordre $d + 1$) sont définies récursivement par l'algorithme Cox-De Boor :

$$N_{i,0}(t) = \begin{cases} 0 & \text{si } t_i \leq t < t_{i+1} \text{ et } t_i < t_{i+1} \\ 1 & \text{sinon} \end{cases} \quad (2.2)$$

$$N_{i,d}(t) = \frac{t - t_i}{t_{i+p} - t_i} N_{i,p-1}(t) + \frac{t_{i+p+1} - t}{t_{i+p+1} - t_{i+1}} N_{i+1,p-1}(t) \quad (2.3)$$

De même, les surfaces NURBS sont définies par deux paramètres (u, v) et une grille de points de contrôle pondérés. Étant donnés deux vecteurs de noeuds $\vec{u} = \{u_0, \dots, u_n\}$ et $\vec{v} = \{v_0, \dots, v_m\}$, deux nombres entiers $p, q > 0$, un ensemble de points de contrôle $P^w = \{P_{ij} | i \in \{0, n-p\}, j \in \{0, m-q\}\}$ et des poids $w_{ij} \in \mathbb{R}$ non nuls. La surface NURBS $S(u, v)$ de degrés (p, q) est définie par l'équation :

$$S(u, v) = \frac{\sum_{i=0}^{n-p} \sum_{j=0}^{m-q} N_{i,p}(u) N_{j,q}(v) w_{ij} P_{ij}}{\sum_{i=0}^{n-p} \sum_{j=0}^{m-q} N_{i,p}(u) N_{j,q}(v) w_{ij}} \quad (2.4)$$

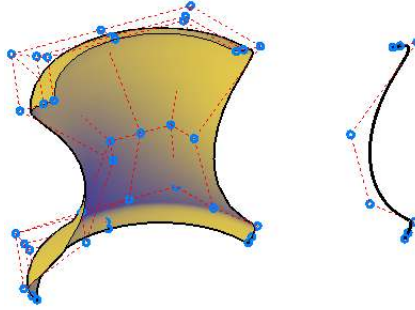


FIGURE 2.7 – Une courbe (à droite) et une surface (à gauche) NURBS avec leurs points de contrôle.

Les surfaces NURBS peuvent être utilisées pour générer des objets 3D de toutes formes : des primitives (droite, cercle, sphère, ellipse) aux formes arbitraires (formes gauches comme une voiture, un animal, etc.).

2.1.2 Pourquoi utiliser les NURBS ?

Piegl [Piegl 1991] a résumé les avantages des courbes et des surfaces NURBS :

1. Les NURBS offrent une forme mathématique unifiée pour générer toutes formes d'objets 3D. Avec le support des points de contrôle et les poids associés, les NURBS permettent aux utilisateurs de générer de façon intuitive une grande variété de formes de façon très flexible.
2. L'évaluation des propriétés géométriques des NURBS est stable et raisonnablement rapide.
3. Grâce aux réseaux de points de contrôle, les NURBS ont une interprétation géométrique claire et adaptée aux utilisateurs qui ont a priori la description géométrique de l'objet à concevoir. De plus, la pondération des points de contrôle des NURBS permet la génération des cercles ou des ellipses plus exacte que des B-Splines.

4. Les NURBS bénéficient d'outils puissants tels que l'insertion/la suppression des noeuds, l'élévation de degré, le découpage. Ces outils permettent l'utilisation facile des NURBS dans la conception, l'analyse, le traitement et l'interrogation des objets 3D.
5. Les NURBS sont invariantes par transformation affine.

Grâce à ses avantages, les NURBS sont largement utilisées dans le domaine CAO avec une communauté d'utilisateurs et d'entreprises très importante. Les NURBS restent néanmoins toujours un sujet de recherche actuel. L'analyse et le traitement des NURBS visant des applications dans la conception et la fabrication industrielles génèrent plusieurs travaux depuis son apparition. Par conséquent, les surfaces NURBS constituent l'objectif de recherche de notre travail.

2.2 Représentation par Bords

Dans certains cas, les objets 3D ne peuvent pas être générés à partir d'une seule surface NURBS : par exemple, des objets ayant plusieurs trous à l'intérieur. Alors, il existe deux méthodes de modélisation très connues supportant la conception des objets 3D de forme arbitraire : CSG (la Géométrie de Construction des Solides) et BRep (la Représentation par Bords). Ces deux méthodes ont été brièvement introduites dans la section précédente. D'après Stroud [Stroud 2011], les BRep sont actuellement plus utilisées que les CSG. En effet, les modèles CSG sont générés en combinant les primitives par des opérateurs booléens. La CSG modélise seulement à partir des primitives comme les sphères, les cubes, les tores et les plans. La génération des objets de forme complexe requiert de nombreuses opérations sur ces primitives. En revanche, les BRep peuvent supporter les surfaces de forme gauche, comme les NURBS, ce qui rendent la conception des objets de forme complexe plus simple. Généralement, que ce soit en représentation CSG ou BRep, la représentation d'un objet 3D n'est pas unique.

La figure 2.8 illustre la spécification de la structure de données d'un modèle BRep. D'après cette spécification, appelée *BRep Théorique*, un modèle BRep dispose d'éléments topologiques et d'éléments géométriques. Les éléments topologiques sont des abstractions des éléments géométriques, où un modèle se décompose d'un ou plusieurs *shells* (des volumes indépendants), eux-mêmes composés de plusieurs faces. Une *face* correspond à une surface découpée par une ou plusieurs *loops*, chaque *loop* est un contour fermé composé d'*arêtes*. Une *arête* est définie par une *courbe* délimitée par un (cas fermé) ou plusieurs (cas ouvert) sommets qui sont tout simplement des points. Pour qu'un modèle BRep soit une variété dans l'espace de dimension 3, c'est-à-dire un modèle de forme volumique fermée, chaque surface doit correspondre à une seule face et chaque courbe doit correspondre à deux arêtes.

La validité des modèles BRep est assurée par la relation *Euler-Poincaré*. Étant donné un modèle BRep valide, on a :

$$v - e + f - r = 2(s - h). \quad (2.5)$$

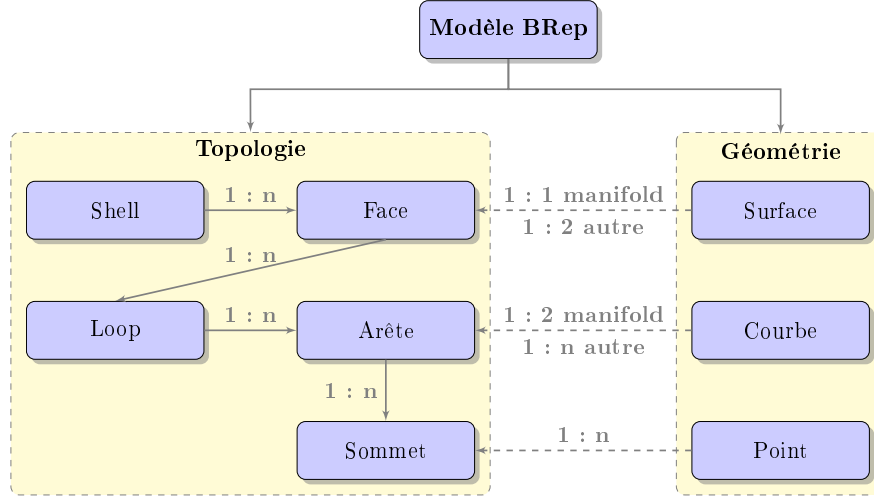


FIGURE 2.8 – Structure de données d'un objet BRep

où $\left\{ \begin{array}{l} v : \text{le nombre de sommets.} \\ e : \text{le nombre d'arêtes.} \\ f : \text{le nombre de faces.} \\ r : \text{le nombre de loops intérieures.} \\ s : \text{le nombre de shells.} \\ h : \text{le nombre de trous.} \end{array} \right.$

Alors, tous les modèles BRep sont identifiables topologiquement comme des *points de paramètres* dans un espace de dimension 6. Ces points se situent sur un hyperplan défini par :

$$v - e + f + 2h - r - 2s = 0. \quad (2.6)$$

En pratique, pour créer un modèle BRep, des opérations « ajouter » et « supprimer » sont effectuées sur les éléments topologiques. Si ces opérateurs s'effectuent en vérifiant la relation *Euler-Poincaré* (équation (2.5)), ils seront appelés *opérateurs d'Euler*. D'après Stroud [Stroud 2011], il existe 99 opérateurs d'Euler possibles qui changent au maximum un élément arbitraire. Tous les changements, toutes les opérations simples ou complexes peuvent être décrites en fonction des compositions de ces opérateurs. Parmi eux, nous pouvons choisir 5 opérateurs indépendants permettant la navigation dans l'hyperplan correspondant à l'équation (2.6). Nous développons ici un exemple pour illustrer l'usage des opérateurs d'Euler. Supposons que M soit l'opération d'ajout, K soit l'opération de suppression, V, E, F, H, R, S soient respectivement des sommets, des *arêtes*, des *faces*, des *trous*, des *loops* et des *shells*. Les 5 opérateurs indépendants choisis et leurs formes vectorielles (v, e, f, h, r, s) sont définis par :

- MEV - $(1, 1, 0, 0, 0, 0)$: ajouter une arête et un sommet.
- MEF - $(0, 1, 1, 0, 0, 0)$: ajouter une arête et une face.
- KEMR - $(0, -1, 0, 0, 1, 0)$: ajouter une loop intérieure, supprimer une arête.

- MVFS - $(1, 0, 1, 0, 0, 1)$: ajouter un sommet, une face et un shell.
- KFMHR - $(0, 0, -1, 1, 1, 0)$: ajouter un trou et une loop intérieure, supprimer une face.

En associant ces opérateurs au vecteur $(1, -1, 1, 2, -1, -2)$ extrait de l'équation (2.6), on peut former une *matrice d'opérations* en ordonnant les six vecteurs en colonnes :

$$A = \begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & -1 & 0 & 0 & -1 \\ 0 & 1 & 0 & 1 & -1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 2 \\ 0 & 0 & 1 & 0 & 1 & -1 \\ 0 & 0 & 0 & 1 & 0 & -2 \end{pmatrix} \quad (2.7)$$

Puisque le nombre des éléments topologiques pour un modèle donné est toujours déterminé, on peut estimer le nombre d'opérateurs d'Euler à effectuer pour générer ce modèle par l'équation :

$$q = Ap. \quad (2.8)$$

où q est le vecteur représentant le nombre d'éléments topologiques et p le nombre d'opérateurs d'Euler à effectuer. De plus, $A^{-1}q = A^{-1}Ap$, alors :

$$p = A^{-1}q \quad \text{avec} \quad A^{-1} = \frac{1}{12} \begin{pmatrix} 9 & 3 & -3 & -6 & 3 & -6 \\ -5 & 5 & 7 & 2 & 5 & -2 \\ 3 & -3 & 3 & -6 & 9 & -6 \\ 2 & -2 & 2 & 4 & -2 & 8 \\ -2 & 2 & -2 & 8 & 2 & 4 \\ 1 & -1 & 1 & 2 & -1 & -2 \end{pmatrix} \quad (2.9)$$

Par exemple, le vecteur d'éléments topologiques d'un cube est $q = (8, 12, 6, 0, 0, 1)$, c'est-à-dire 8 sommets, 12 arêtes, 6 faces et 1 shell. Appliquons l'équation (2.9), nous obtenons $p = (7, 5, 0, 1, 0, 0)$. C'est-à-dire, pour générer un modèle BRep d'un cube, il faut effectuer successivement 7 MEV, 5 MEF et 1 MVFS.

Nous constatons que, par rapport aux opérateurs booléens de la CSG, la BRep admet plus d'opérations pour générer des formes 3D, elle donne plus de libertés aux utilisateurs pour générer les formes 3D arbitraires. C'est l'une des raisons pour lesquelles les modèles BRep sont davantage présents dans le domaine de la CAO que les modèles CSG.

Dans notre travail, nous nous intéressons aux modèles BRep définis par des surfaces et des courbes NURBS. Comme nous l'avons mentionné, les modèles BRep basés NURBS (BRep-NURBS) constituent un standard dans la conception et la fabrication de l'industrie. Nous utilisons la librairie *OpenNURBS*² de *Mc Neel & Associates*, pour l'évaluation et le traitement des modèles BRep-NURBS. Dans la suite, nous introduisons la spécification des modèles BRep dans le contexte d'OpenNURBS.

2. <http://wiki.mcneel.com/developer/opennurbs/home>

2.3 BRep dans le contexte d'OpenNURBS

OpenNURBS est une librairie, ouverte et sans condition d'utilisation, qui offre des outils permettant la lecture et l'écriture des modèles 3DM (une spécification de *Mc Neel & Associates*), soit polygonaux, soit surfaciques. Dans notre travail, nous considérons uniquement des modèles surfaciques BRep basés NURBS. La structure d'un modèle BRep-NURBS dans le contexte d'OpenNURBS est illustrée dans la figure 2.9a. Dans cette structure, nous pouvons retrouver l'équivalence entre la spécification de BRep Théorique (figure 2.8) et celle d'OpenNURBS. Comme le tableau 2.1 le montre, un modèle 3DM contient plusieurs éléments BRep (shell) qui se composent de plusieurs *faces*, une *face* est définie par une surface découpée par une ou plusieurs *loops*. Il existe en plus l'élément *trim* qui est lié à un *edge* (arête) défini par une courbe NURBS en 3D. Par convenance, dans le reste de ce manuscrit, nous utilisons les concepts en anglais pour indiquer les éléments topologiques des modèles BRep suivants la spécification d'OpenNURBS.

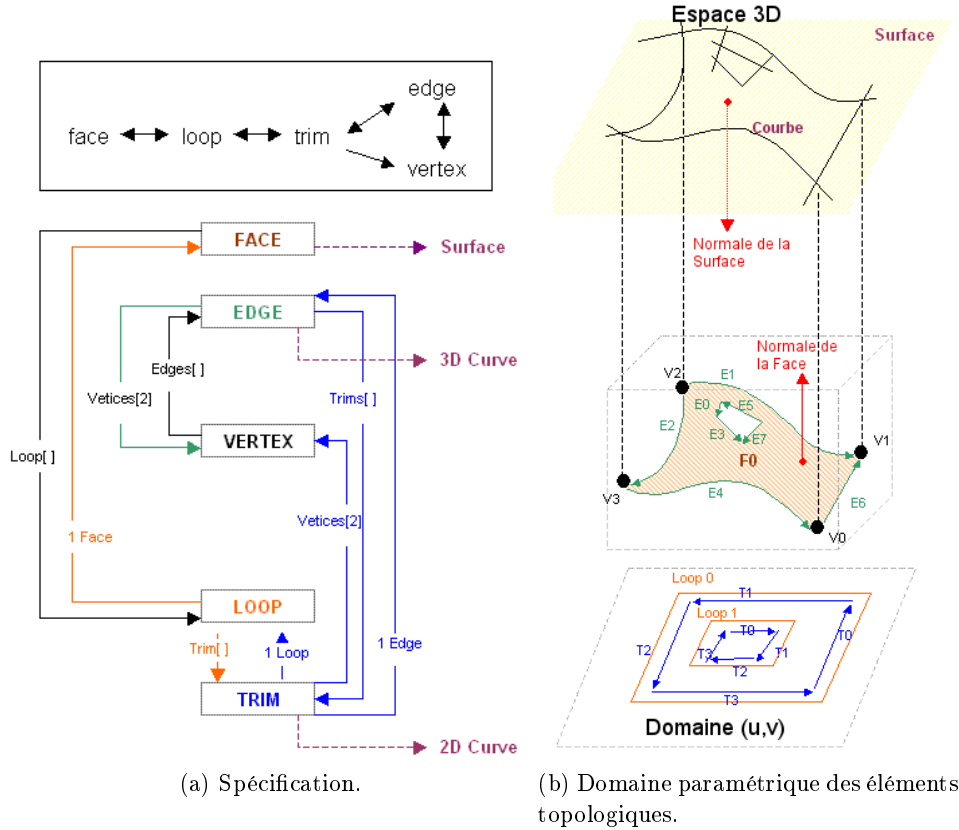


FIGURE 2.9 – Représentation par Bords dans le contexte d'OpenNURBS.³

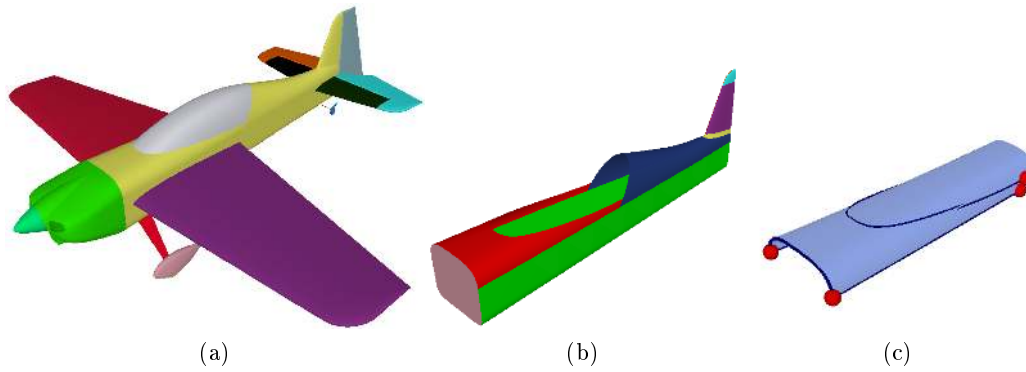


FIGURE 2.10 – Exemple d'un modèle BRep d'un avion.

OpenNURBS	BRep Théorique	Dimension
BRep	Shell	3D
Face	Face	3D
Loop	Loop	2D
Trim	-	2D
Edge	Arête	3D
Vertex	Sommet	3D

TABLE 2.1 – Équivalence entre les spécifications d'OpenNURBS et de BRep Théorique.

En raison des erreurs numériques dans le calcul paramétrique, il existe une tolérance dans l'approximation de la courbe NURBS (de l'objet *edge*) à la courbe réelle sur la surface. L'objet *trim* consiste en une courbe NURBS en 2D qui définit le domaine (u, v) de la courbe réelle située sur la surface. La figure 2.9b illustre le domaine (u, v) des éléments *trims* d'une *face*. De plus, le sens des *trims* définit la partie à découper d'une surface. En se référant par rapport à la normale de la *face*, les parties à droite des *trims* sont considérées extérieures et ne sont pas rattachées au modèle. Pour les *loops*, il existe deux types : intérieur et extérieur. Pour qu'un modèle soit topologiquement valide, les *loops* ne doivent pas s'intersecter, une *loop* ne peut pas contenir une autre *loop* de même type et un modèle doit posséder seulement une *loop* extérieure.

La figure 2.10 illustre l'exemple d'un modèle BRep-NURBS d'un avion suivant la spécification d'OpenNURBS. Le modèle se compose de plusieurs BReps décrites en différentes couleurs sur la figure 2.10a, le BRep représentant le fuselage en jaune contient plusieurs *faces* décrites en différentes couleurs sur la figure 2.10b. Chaque *face* est une surface NURBS découpée par des *trims* eux-mêmes limités par des Vertex. La figure 2.10c détaille les éléments géométriques de la Face rouge de la figure 2.10b : la surface NURBS en bleu, les points en rouge et les courbes en bleu foncé.

3. <http://wiki.mcneel.com/developer/brepstructure>

Dans la plupart des traitements géométriques, l'échantillonnage des points situés sur la surface est requise. Pour les surfaces NURBS qui sont définies sur leur domaine des paramètres (u, v) , l'accès à ces échantillons est facile en naviguant dans ces domaines. Pour les modèles BRep-NURBS, l'échantillonnage des *faces* est plus compliqué puisque les échantillons sont délimités par des courbes *trim*. Il faut alors des méthodes spécifiques pour échantillonner ces *faces*. Dans le paragraphe suivant, nous introduisons nos algorithmes d'échantillonnage.

2.4 Échantillonnage des Faces

Étant donné une *face* F découpée d'une surface NURBS $S(u, v)$ définie par l'équation (2.4), les échantillons peuvent être récupérés en naviguant le long des domaines de paramètres u et v de la surface NURBS. L'échantillonnage des surfaces paramétriques génère beaucoup de travaux dans l'inspection des surfaces de forme gauche [Elkott 2002, Obeidat 2009]. D'après Elkott, il existe deux méthodes connues pour échantillonner les surfaces paramétriques :

- **Équi-paramétrique** : les échantillons sont distribués régulièrement dans l'espace (u, v) . Cette méthode est appropriée pour les surfaces simples dont la variation de leurs courbures n'est pas importante [Obeidat 2009]. Supposons que $u \in [u_{min}, u_{max}]$, $v \in [v_{min}, v_{max}]$ et que nous voulons prendre respectivement n_u, n_v échantillons dans les sens des vecteurs des noeuds \vec{u}, \vec{v} , cette méthode est présentée dans l'algorithme 1.
- **Basée sur la taille des patches** : la surface est divisée en plusieurs patches suivant ses vecteurs de noeuds. Ces patches sont ordonnés en fonction de leurs tailles. Le nombre d'échantillons est défini à partir du rang de chaque patch, le patch plus grand possède un nombre d'échantillons plus grand et proportionnel à sa taille.

Nous constatons que la méthode *équi-paramétrique* donne une grille régulière des points où chaque point possède 8 points voisins autour. Par contre, la méthode *basée sur la taille des patches* donne une grille irrégulière de points. Dans notre algorithme de détection de similarités dans le chapitre suivant, il est nécessaire de maintenir un ensemble de points facilitant l'identification des voisins d'un point donné. Malgré les résultats intéressants dans les travaux [Elkott 2002, Obeidat 2009], leurs échantillons générés ne satisfont pas la facilité de navigation exigée. Alors, nous choisissons la méthode *équi-paramétrique* pour échantillonner des faces dans notre travail.

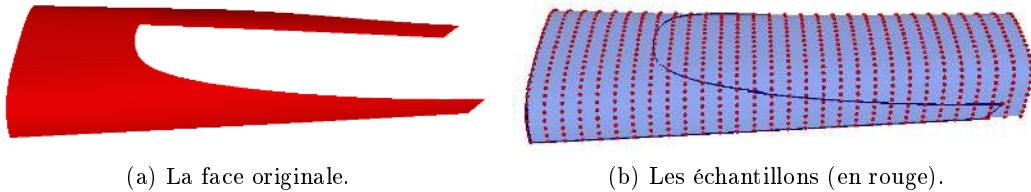


FIGURE 2.11 – Échantillonnage équi-paramétrique d'une face.

Algorithme 1 Algorithme d'échantillonnage équi-paramétrique.

Précondition : $n_u > 1, n_v > 1$.

$L = \{\emptyset\}$ %Liste des échantillons

$$du = \frac{u_{max} - u_{min}}{n_u - 1}, dv = \frac{v_{max} - v_{min}}{n_v - 1}.$$

pour $i = 0$ à $n_u - 1$ **faire**

$$u = u_{min} + i \times du$$

pour $j = 0$ à $n_v - 1$ **faire**

$$v = v_{min} + j \times dv$$

$$p = S(u, v)$$

Ajouter p à la liste L

retour L

En reprenant l'exemple d'une face du fuselage de l'avion de la figure 2.10c, les échantillons extraits en appliquant cet algorithme sont représentés sur la figure 2.11b. Nous constatons qu'il existe deux problèmes :

- Les échantillons appartiennent non seulement à l'intérieur de la face mais aussi à la région découpée par les bords. Car les échantillons sont extraits sur toute la surface sous-jacente sans prendre en compte un mécanisme de vérifier les positions à l'intérieur de la face.
- La distance euclidienne entre les échantillons n'est pas uniforme, alors que des échantillons distribués régulièrement sur une surface 3D rendent l'analyse et le traitement géométrique plus robustes. En effet, dans une surface paramétrique, il n'est pas tout à fait certain que des points espacés régulièrement dans l'espace paramétrique soient aussi espacés régulièrement dans l'espace géométrique.

Alors, l'échantillonnage équi-paramétrique n'est pas une bonne approche pour les algorithmes analysant ses caractéristiques géométriques. Par définition, une face consiste aussi en des sommets qui sont des points paramétrés sur la surface NURBS. Les paramètres (u, v) des sommets définissent une région paramétrique fermée qui enveloppe les points de la face. Dans la suite, nous introduisons nos méthodes d'échantillonnage pour résoudre les problèmes de l'échantillonnage équi-paramétrique.

2.4.1 Échantillonnage à l'intérieur de la face

Grâce au support des sommets, cet algorithme d'échantillonnage nous permet de prendre les points situés à l'intérieur de la face, bornés par les arêtes de cette face. Comme nous l'avons mentionné, les paramètres (u, v) des arêtes d'une face définissent une région paramétrique fermée que nous appelons le **domaine paramétrique** de cette face. Ce domaine est utilisé comme une référence pour identifier les points de cette face. Nous commençons par le cas où les arêtes sont polygonaux : les paramètres des sommets forment un polygone en 2D dont les sommets sont ordonnés a priori suivant une direction arbitraire. Cet ordre peut être relevé à partir des *loops* de la face. En d'autres termes, le domaine paramétrique d'une face est

déterminé à partir des régions paramétriques englobées par ses *loops*.

Soit une face d'un objet BRep, nous posons la **région paramétrique** d'une *loop* L de cette face, notée $Region(L)$, comme l'enveloppe (convexe ou concave) fermée définie par les paramètres (u, v) des sommets ordonnés de cette *loop*.

Dans le contexte d'OpenNURBS, les paramètres (u, v) des sommets d'une *loop* sont relevés depuis les courbes 2D (voir figure 2.9a) attachées à l'objet BRep. En effet, ces courbes définissent les domaines paramétriques des arêtes ainsi que des sommets dans le domaine paramétrique de la surface NURBS sous-jacente. Tous les trims ont deux extrémités et sont ordonnés les uns après les autres pour former un contour fermé d'une *loop*. La région paramétrique est définie de façon exacte pour les faces dont les arêtes sont droites dans l'espace 2D. Pourtant, il existe aussi des arêtes courbées (voir figure 2.10c), l'utilisation des deux sommets de ces arêtes n'est pas suffisante pour définir la région paramétrique. Dans ce travail, en dehors de deux sommets d'un trim, nous échantillons régulièrement un certain nombre de points le long de la courbe 2D attachée pour raffiner la région paramétrique.

Rappelons qu'une face F d'un objet BRep se compose d'une seule *loop* extérieure L^{ext} et peut contenir n_L *loops* intérieures L_i^{int} ($n_L \geq 0$ et $i \in [0, n_L - 1]$), ces *loops* ne s'intersectent pas. Nous profitons de cette structure pour définir le domaine paramétrique d'une face et la condition pour qu'un échantillon soit dans cette face.

Définition 2 Le **domaine paramétrique** \mathbb{D}^F de la face F est défini par l'équation :

$$\mathbb{D}^F = Region(L^{ext}) \setminus (\oplus Region(L_i^{int})_{0 \leq i \leq n_L - 1}). \quad (2.10)$$

Proposition 1 Un échantillon est dit à l'intérieur de la face F si et seulement si ses paramètres correspondants $(u, v) \in \mathbb{D}^F$.

Grâce au support du domaine paramétrique, l'algorithme 1 est amélioré pour assurer que les échantillons soient dans une face F donnée.

Algorithme 2 Algorithme d'échantillonnage à l'intérieur d'une face.

Précondition : $n_u > 1, n_v > 1$.

Déterminer le domaine paramétrique \mathbb{D}^F .

$E = \{\emptyset\}$ %Liste des échantillons

$du = \frac{u_{max} - u_{min}}{n_u - 1}, dv = \frac{v_{max} - v_{min}}{n_v - 1}$.

pour $i = 0$ à $n_u - 1$ **faire**

$u = u_{min} + i \times du$

pour $j = 0$ à $n_v - 1$ **faire**

$v = v_{min} + j \times dv$

si $(u, v) \in \mathbb{D}^F$ **alors**

$p = S(u, v)$

Ajouter p à la liste E

retour E

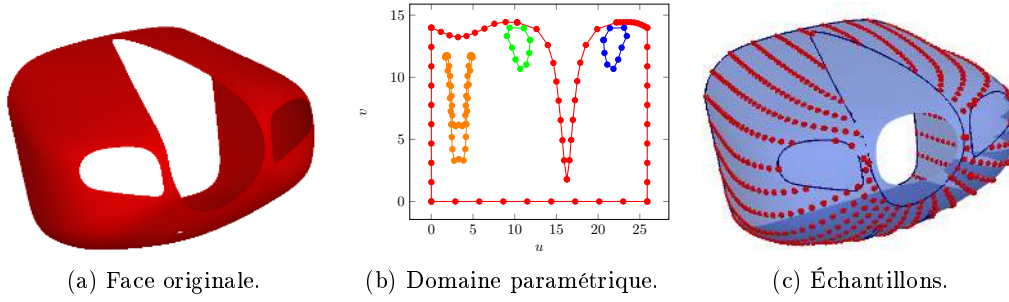


FIGURE 2.12 – Résultat de l'échantillonnage d'une face.

Le résultat de l'algorithme d'échantillonnage à l'intérieur de la *face* est illustré par la figure 2.12. Dans cette figure, la *face* a une *loop* extérieure et trois *loops* intérieures. Nous constatons que les échantillons sont correctement relevés.

2.4.2 Échantillonnage de distance relativement uniforme

Dans l'algorithme d'échantillonnage équi-paramétrique, il existe une opération qui produit une **grille de points paramétriques**, notée $Grille(F)$, à partir de laquelle les échantillons de la face F sont évalués :

$$Grille(F) = \{(u^i, v^j) \mid i \in [0, n_u - 1], j \in [0, n_v - 1]\}. \quad (2.11)$$

Où $u^i \in [u_{min}, u_{max}]$, $v^j \in [v_{min}, v_{max}]$, n_u et n_v sont les nombres d'échantillons dans les directions u et v de la face F . Supposons $du = \frac{u_{max} - u_{min}}{n_u - 1}$ et $dv = \frac{v_{max} - v_{min}}{n_v - 1}$, nous avons la condition pour que les points paramétriques soient dans la grille :

$$\begin{cases} |u^i - u^{i+1}| &= du. \\ |v^j - v^{j+1}| &= dv. \end{cases}$$

Comme nous l'avons mentionné, il n'est pas certain que des points espacés régulièrement dans l'espace paramétrique soient aussi espacés régulièrement dans l'espace géométrique. Pour que les échantillons soient espacés régulièrement sur la face F , $Grille(F)$ doit être redéfinie : c'est l'idée principale de cet algorithme. Étant donnée une distance minimum de entre les échantillons, la génération de la grille est décrite par l'algorithme 3. Cet algorithme détermine *une séquence de référence* des paramètres, dans chacune des directions u et v , à partir desquels les échantillons sont espacés de façon relativement uniforme. En utilisant ces deux séquences, la grille paramétrique est générée pour l'échantillonnage.

La figure 2.13 présente les résultats de l'échantillonnage en combinant cet algorithme avec l'échantillonnage à l'intérieur de la face, les échantillons sont distribués régulièrement et sont situés sur les faces.

Algorithme 3 Algorithme de génération de la grille paramétrique.

Précondition : $de > 0$.

$G = \{\emptyset\}$ %Grille des points paramétriques.

$L^u = \{\emptyset\}, L^v = \{\emptyset\}$ %Listes des paramètres u et v .

$du = \frac{u_{max} - u_{min}}{n_u - 1}, dv = \frac{v_{max} - v_{min}}{n_v - 1}.$

$u = u_{min}, v = \frac{v_{max} + v_{min}}{2}.$

tant que $u < u_{max}$ **faire**

tant que $\|S(u, v) - S(u + du, v)\| > de$ **faire**

 Régler du .

 Ajouter u à L^u .

$u = u + du$.

$v = v_{min}, u = \frac{u_{max} + u_{min}}{2}.$

tant que $v < v_{max}$ **faire**

tant que $\|S(u, v) - S(u, v + dv)\| > de$ **faire**

 Régler dv .

 Ajouter v à L^v .

$v = v + dv$.

pour tout u dans L^u **faire**

pour tout v dans L^v **faire**

 Ajouter (u, v) à G .

retour G

2.4.3 Échantillonnage par pondération

L'opération commune des algorithmes d'échantillonnage précédents est la navigation le long des directions des paramètres u et v . Dans ce paragraphe, nous introduisons un autre algorithme d'échantillonnage qui se base sur les sommets des faces. L'idée principale de cet algorithme est l'application du calcul de coordonnée barycentrique sur les triplets de sommets. Étant donnés trois points A , B et C

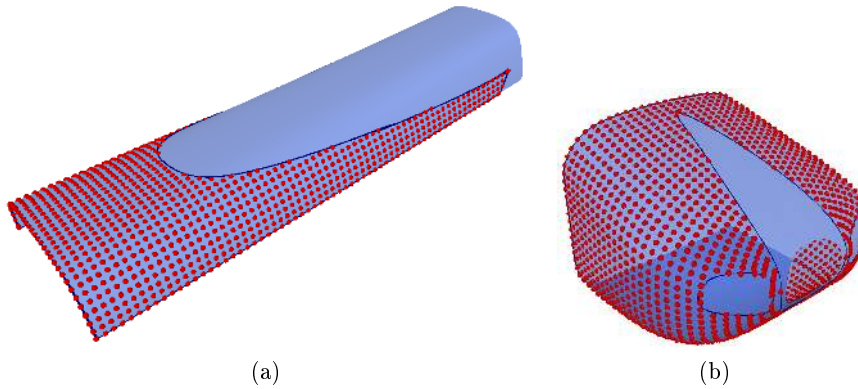


FIGURE 2.13 – Résultats de l'échantillonnage de distance relative uniforme.

dans un espace réel (figure 2.14), tous les points $P \in \Delta ABC$ sont déterminés par l'équation :

$$P = tA + rB + (1 - r - t)C. \quad (2.12)$$

Avec $r + t \leq 1$ et $r, t \in [0, 1]$, l'échantillonnage est donc effectué en variant les paramètres r et t dans l'intervalle $[0, 1]$.

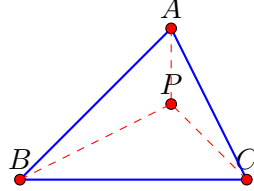


FIGURE 2.14 – Illustration du calcul de coordonnée barycentrique.

Pour les faces ayant plus de trois sommets, nous utilisons la *Triangulation de Delaunay contrainte* [Paul Chew 1989] pour diviser la région paramétrique de la loop extérieure en plusieurs triangles. À partir de ces triangles, les échantillons sont évalués. L'algorithme 5 décrit cette approche d'échantillonnage d'une face associée avec le domaine paramétrique de cette face.

Algorithme 4 Algorithme d'échantillonnage par pondération.

Précondition : $n \geq 3$ %Nombre de sommets de la face.

$E = \emptyset$ %Liste des échantillons.

Évaluer la région paramétrique R de la loop extérieure et le domaine \mathbb{D}^F .

$\Delta = \text{Triangulation}(R)$ %Triangulation de la région R .

pour tout Triangle $ABC \in \Delta$ **faire**

 Pondérer (u, v) à partir de trois points A, B, C .

si $(u, v) \in \mathbb{D}^F$ **alors**

$p = S(u, v)$

 Ajouter p à la liste E

retour E

2.4.4 Conclusion

Dans ce travail, nous introduisons trois algorithmes d'échantillonnage d'une face du modèle BRep-NURBS pour résoudre le problème d'échantillons en dehors de la face et la régularité spatiale. En se basant sur *le domaine paramétrique* d'une face, les échantillons sont facilement générés à l'intérieur de cette face. De plus, en utilisant une distance minimum entre les échantillons, la *grille des points paramétriques* d'une face est plus régulière dans l'espace paramétrique. Cette grille est utilisée pour générer des échantillons espacés de distances relativement uniformes sur la face. Même si les résultats obtenus restent dépendants de la paramétrisation des surfaces, ces résultats sont en pratique corrects et sont suffisants pour notre algorithme de détection des similarités. L'échantillonnage par pondération est une

approche d'échantillonnage indépendante des directions des paramètres u et v des faces du modèle BRep-NURBS. Nous l'utilisons pour la comparaison entre deux faces dont les sommets sont mis en correspondance. Cette utilisation sera détaillée dans le chapitre 3.

En outre, nous utilisons l'algorithme d'échantillonnage à l'intérieur de la face pour évaluer la **boîte englobante** de cette face. Étant donné une face F et l'ensemble des échantillons générés par cet algorithme $E = \{e^i = (e_x^i, e_y^i, e_z^i) | i \in [0, N^e - 1]\}$. La boîte englobante B^F est définie par deux sommets $s_{min} = (x_{min}, y_{min}, z_{min})$ et $s_{max} = (x_{max}, y_{max}, z_{max})$:

$$\begin{cases} x_{min} = \min e_x^i, & y_{min} = \min e_y^i, & z_{min} = \min e_z^i \\ x_{max} = \max e_x^i, & y_{max} = \max e_y^i, & z_{max} = \max e_z^i \end{cases} \quad (2.13)$$

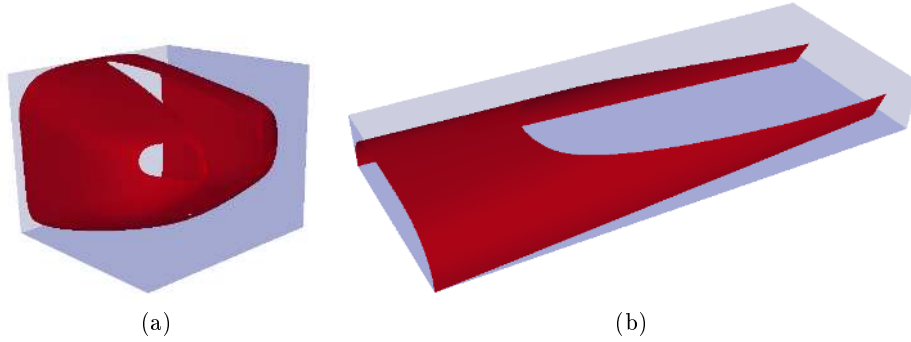


FIGURE 2.15 – Les boîtes englobantes de deux faces.

En fait, cette boîte englobante est alignée avec les axes. La figure 2.15 présente les boîtes englobantes évaluées pour deux faces. Ces boîtes englobantes seront, par la suite, utilisées dans les algorithmes que nous allons proposer.

Dans ce travail de recherche, notre intérêt est la détection des similarités partielles dans des modèles BRep-NURBS. En profitant de la structure des BReps et en héritant des travaux précédents, nous proposons d'utiliser l'**isométrie** dans l'espace Affine Euclidien pour caractériser les similarités. Dans la section suivante, nous allons introduire les bases théoriques concernant les isométries dans l'espace tridimensionnel. En particulier, la classification et une métrique des isométries sont des facteurs importants notre algorithme de détection.

3 Isométrie : caractérisation de similarité

Dans cette partie, les théories principales de la *Géométrie Affine* liées à ce travail sont présentées en détails. Les notions de base dans l'espace Affine Euclidien de dimension 3 sont tout d'abord rappelées. Ensuite, le groupe orthogonal des isométries linéaires dans l'espace vectoriel de dimension 3 est expliqué avant d'introduire des isométries dans l'espace Affine Euclidien. Ces théories sont synthétisées depuis les

ouvrages *Géométrie Affine, Projective et Euclidienne* de Tisseron [Tisseron 1988] et *Méthodes modernes en Géométrie* de Fresnel [Fresnel 2010].

3.1 Isométries dans l'espace tridimensionnel

Soient un ensemble V non vide sur un corps \mathbb{K} (corps réel \mathbb{R} ou corps complexe \mathbb{C}) et deux lois « $+$ » et « \cdot ». Le neutre pour $+$ est $0_{\mathbb{K}}$, le neutre pour \cdot est $1_{\mathbb{K}}$.

Définition 3 On note $(V, +, \cdot)$ l'ensemble V muni d'une loi interne $+$ (opération entre les éléments de V) et d'une loi externe \cdot (opération entre un élément de V et un élément du corps \mathbb{K}). On dit que V est un **espace vectoriel** lorsque :

- $(V, +)$ forme un groupe commutatif : la loi interne est associative et possède un élément neutre, tout élément de V a un inverse.
- $\forall \lambda \in \mathbb{K}, \forall (x, y) \in V^2, \lambda \cdot (x + y) = \lambda \cdot x + \lambda \cdot y$.
- $\forall (\lambda, \mu) \in \mathbb{K}^2, \forall x \in V, (\lambda + \mu) \cdot x = \lambda \cdot x + \mu \cdot x$.
- $\forall (\lambda, \mu) \in \mathbb{K}^2, \forall x \in V, (\lambda \cdot \mu) \cdot x = \lambda \cdot (\mu \cdot x)$.
- $\forall x \in V, 1_{\mathbb{K}} \cdot x = x$.

Les éléments de V sont appelés des **vecteurs** et les éléments de \mathbb{K} des **scalaires**. De plus, il existe un **vecteur nul**, noté 0_V , tel que $\forall x \in V, 0_{\mathbb{K}} \cdot x = 0_V$.

Pour la cohérence dans la suite, nous supposons un espace vectoriel V de dimension 3 sur \mathbb{R} muni d'un produit scalaire $\langle \cdot | \cdot \rangle$. Pour $v \in V$, on note $\langle v | v \rangle = \|v\|^2$, $\|v\|$ est appelée la norme de V . Pour commencer, les notions de base concernant les isométries dans l'espace Affine Euclidien sont rappelées.

Définition 4 Soit une base \mathcal{B} définie dans l'espace vectoriel V de dimension 3 et admettant des éléments notés e_i , avec $i \in \{1, 2, 3\}$, alors \mathcal{B} est une **base orthonormée** si :

- $\forall i, \|e_i\| = 1$.
- $\forall i, j$ et $i \neq j, \langle e_i | e_j \rangle = 0$.

En notant $x = \sum_{i=1}^3 x_i e_i$ et $y = \sum_{i=1}^3 y_i e_i, \forall x, y \in V$, le produit scalaire peut s'écrire :

$$\langle x | y \rangle = \sum_{i=1}^3 x_i y_i.$$

Définition 5 Une application linéaire $\vec{f} : V \rightarrow V$ est dite **orthogonale** si elle préserve le produit scalaire, c'est-à-dire, $\forall x, y \in V, \langle \vec{f}(x) | \vec{f}(y) \rangle = \langle x | y \rangle$. L'application \vec{f} est aussi appelée **isométrie linéaire**.

L'ensemble de toutes les isométries linéaires dans V est noté $\mathcal{O}(V)$. Dans ce cas, l'isométrie \vec{f} peut être représentée dans la base orthonormée \mathcal{B} sous forme matricielle $M(\vec{f}, \mathcal{B}) = A$ telle que $\vec{f}(v) = Av$, avec la matrice $A = [a_{ij}]_{3 \times 3}, a_{ij} \in \mathbb{R}$ et $\det(\vec{f}) = \det(A) \neq 0$. Alors, nous avons les conditions équivalentes suivantes :

- \vec{f} est orthogonale.

- \vec{f} préserve le produit scalaire, $\forall x, y \in V, \langle x|y \rangle = \langle \vec{f}(x)|\vec{f}(y) \rangle$.
- A est telle que $A^t A = I_3$, avec I_3 la matrice identité d'ordre 3, soit $A^{-1} = A^t$. Alors, on dit que A est une **matrice orthogonale**, et l'ensemble de ces matrices est noté $\mathcal{O}_3(\mathbb{R})$.
- Le déterminant de A vaut 1 ou -1 .

Définition 6 Un ensemble non vide E est un **espace affine** de direction V s'il existe une application de $E \times E$ dans V notée $(x, y) \mapsto \vec{xy}$ vérifiant les conditions suivantes :

- $\forall x, y, z \in E, \vec{xy} + \vec{yz} = \vec{xz}$ (relation de Chasles).
- $\forall x \in E, \vec{v} \in V, \exists y \in E, \vec{xy} = \vec{v}$.

Définition 7 Un espace affine E de direction V est dit **euclidien** si V est euclidien. En d'autre terme, V est un espace vectoriel, de dimension 3, muni d'un produit scalaire $\langle \cdot | \cdot \rangle$.

Dans notre travail, la direction euclidienne V de l'espace affine E sera notée \vec{E} . Pour abréviation, l'espace affine E associé à la direction \vec{E} sera noté (E, \vec{E}) et nous posons E comme un espace affine euclidien.

Définition 8 Une application $f : E \rightarrow E$ est appelée affine s'il existe une application linéaire $\vec{f} : \vec{E} \rightarrow \vec{E}$ satisfaisant $\vec{f}(\vec{xy}) = \vec{f(x)f(y)}$ pour tous $x, y \in E$.

Définition 9 Soit une application affine $f : E \rightarrow E$, f est appelée une **isométrie** si elle conserve la distance $(x, y) \mapsto xy = \|\vec{xy}\|$ sur E , c'est-à-dire $\|\vec{xy}\| = \|\vec{f(x)f(y)}\|$ pour tout $x, y \in E$.

Proposition 2 Pour une application $f : E \rightarrow E$, f est une isométrie ssi f est affine et $\vec{f} \in \mathcal{O}(\vec{E})$.

D'après cette proposition, la partie linéaire d'une isométrie est une application orthogonale. En d'autres termes, une isométrie est une application affine conservant les angles et les distances. En effet, chaque application affine f est associée à une application linéaire \vec{f} . Pour caractériser les isométries, il faut analyser les applications linéaires associées. Ces analyses sont introduites dans les sections suivantes. Tout d'abord, nous introduisons la caractérisation des *isométries linéaires* dans l'espace vectoriel Euclidien. Ensuite, cette dernière est développée pour caractériser les *isométries* dans l'espace Affine Euclidien. Ces notions sont nécessaires, puisqu'elles donnent des critères importants pour classer les différentes isométries.

3.1.1 Isométries linéaires dans l'espace vectoriel Euclidien

Proposition 3 Soit une application linéaire orthogonale $\vec{f} \in \mathcal{O}(\vec{E})$, alors les espaces propres de \vec{f} sont supplémentaires et orthogonaux, les valeurs propres appartiennent à $\{1, -1\}$.

Proposition 4 Toute application orthogonale $\vec{f} \in \mathcal{O}(\vec{E})$ est le produit de symétries planaires. Si $n = \dim \vec{E} \geq 2$, on peut toujours supposer que ce produit contient $p \leq n$ symétries. En effet, si $\vec{f} \neq I_{\vec{E}}$ et $s = \dim \text{Ker}(\vec{f} - I_{\vec{E}})$, alors \vec{f} est un produit de $p = n - s$ symétries planaires.

De plus, si p est pair, \vec{f} est une application préservant l'orientation et l'ensemble de ces applications est noté $\mathcal{O}^+(\vec{E})$. En revanche, si \vec{f} ne préserve pas l'orientation, $\vec{f} \in \mathcal{O}^-(\vec{E})$.

Proposition 5 Soit $\vec{f} \in \mathcal{O}(\vec{E})$, alors :

1. Il existe des sous-espaces orthogonaux deux à deux et invariants par \vec{f} tels que $E = \bigoplus (E_k)_{1 \leq k \leq m}$. De plus, $\forall k \in \mathbb{N}_m$, E_k est de dimension 1 ou 2. L'application induite $f_k : E_k \rightarrow E_k$ est l'**identité** ou son **opposée** si $\dim E_k = 1$, une **rotation** autre que l'identité ou son **opposée** si $\dim E_k = 2$.
2. Il existe une base orthonormée $\mathcal{B} = (e_1, \dots, e_3)$ de \vec{E} pour que la forme matricielle de \vec{f} dans \vec{E} soit telle que :

$$M(\vec{f}, \mathcal{B}) = \begin{bmatrix} I_{p^+} & \cdots & 0 \\ \vdots & -I_{p^-} & \vdots \\ 0 & \cdots & A \end{bmatrix}$$

$$\text{Avec } \begin{cases} p^+ &= \dim E(1, \vec{f}) \\ p^- &= \dim E(-1, \vec{f}) \\ A &= \begin{cases} \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} & \text{si } p^+ + p^- = 1 \\ \square & \text{si non.} \end{cases} \\ \theta &\in]0, 2\pi[\setminus \pi \\ p^+ + p^- + \dim A &= 3. \end{cases}$$

Ces propositions nous permettent de définir un outil pour caractériser une *isométrie linéaire* donnée dans l'espace vectoriel de dimension 3 en considérant les espaces propres associés à cette isométrie. Dans la suite, en analysant la forme matricielle et les espaces propres, les isométries linéaires sont classifiées en groupes et caractérisées par une ou plusieurs propriétés invariantes.

Classe	Espaces propres	Forme matricielle	Caractéristiques	Description
ID	$p^+ = 3, p^- = 0$	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	\emptyset	\vec{f} est l'identité.
SYM	$p^+ = 2, p^- = 1$	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix}$	$P = \text{Ker}(\vec{f} - I_3)$	\vec{f} est une symétrie orthogonale d'hyperplan P .
ROT	$p^+ = 1, p^- = 0$	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{bmatrix}$	$\vec{D} \in \text{Ker}(\vec{f} - I_3)$ $\theta \in]0, \pi[$	\vec{f} est une rotation d'axe orienté par \vec{D} d'une mesure d'angle θ .
RETO	$p^+ = 1, p^- = 2$	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix}$	$\vec{D} \in \text{Ker}(\vec{f} - I_3)$	\vec{f} est un retournement d'axe orienté par \vec{D} .
SYRO	$p^+ = 0, p^- = 1$	$\begin{bmatrix} -1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{bmatrix}$	$\vec{D} \in \text{Ker}(\vec{f} + I_3)$ $\theta \in]0, \pi[$	\vec{f} est une symétrie-rotation dont la symétrie orthogonale d'hyperplan $P = (\vec{D})^\perp$ et la rotation d'axe orienté par \vec{D} d'une mesure d'angle θ .
HOM	$p^+ = 0, p^- = 3$	$\begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix}$	\emptyset	\vec{f} est une homothétie de rapport -1 (une symétrie centrale).

TABLE 2.2 – Groupes d'isométries dans l'espace vectoriel de dimension 3.

On remarque dans cette classification qu'il existe toujours une *base orthonormée* $\mathcal{B} = (e_1, e_2, e_3)$ bien orientée pour que les formes matricielles soient conformes à celles du tableau 2.2. De plus, les isométries linéaires sont aussi classifiées en deux groupes en fonction des déterminants de leurs formes matricielles (dans n'importe quelle base) :

- *Groupe des isométries directes* dont $\det \vec{f} = 1$.
- *Groupe des isométries indirectes* dont $\det \vec{f} = -1$.

3.1.2 Isométries affines dans l'espace Affine Euclidien

Maintenant, nous allons caractériser des isométries dans l'espace affine euclidien.

Définition 10 Soit une application affine $f : E \rightarrow E$, alors $x \in E$ est un point fixe de f si :

$$x = f(x). \quad (2.14)$$

On note :

$$\begin{aligned} E(1, f) & \text{ l'ensemble des points fixes de } f \\ \vec{E}(1, \vec{f}) & \text{ le sous-espace invariant de } \vec{f} \\ & \text{ (ayant pour direction } \text{Ker}(\vec{f} - I_3)) \end{aligned}$$

$\vec{E}(1, \vec{f}) \neq \{o\}$ si et seulement si 1 est la valeur propre de \vec{f} , les éléments non nuls de $\vec{E}(1, \vec{f})$ sont alors des vecteurs propres associés à la valeur propre 1 de \vec{f} .

Dans la suite, nous introduisons des notations des groupes d'isométries suivant la nature des applications linéaires associées :

- $Is(E)$ le groupe des isométries affines.
- $Is^+(E)$ le groupe des isométries directes f telles que $\det \vec{f} = 1$.
- $Is^-(E)$ le groupe des isométries indirectes f telles que $\det \vec{f} = -1$.

Dans notre travail, toutes les isométries sont décomposées en transformations canoniques : translations, rotations axiales et réflexions. La proposition suivante introduit le premier pas de cette décomposition.

Proposition 6 Soit $f \in Is(E)$, alors il existe un couple unique $(g, t_{\vec{a}})$ où g est une isométrie ayant un ensemble non vide de points fixes $G = E(1, g)$, l'ensemble de vecteurs propres associés à la valeur propre 1 de g , et où $t_{\vec{a}}$ est une translation de vecteur $\vec{a} \in \vec{G}$ telles que $f = t_{\vec{a}} \circ g$. On a de plus $f = g \circ t_{\vec{a}}$ et $\vec{G} = \vec{E}(1, \vec{f})$.

1. $f = g$ et $\vec{a} = 0$ si et seulement si f a au moins un point fixe.
2. Si f n'a pas de point fixe, on a $\dim G \geq 1$.

En pratique, lorsque f est donnée, on peut déterminer facilement $\vec{E}(1, \vec{f})$ qui est le sous-espace propre de \vec{f} associé à la valeur propre 1 ($\text{ker}(\vec{f} - \text{Id}_{\vec{E}})$). De plus, étant donné $y \in E$, $t_{\vec{a}}$ est la projection de $\overrightarrow{yf(y)}$ sur $\vec{E}(1, \vec{f})$.

On constate que les isométries linéaires dans l'espace vectoriel sont caractérisées par des sous-espaces vectoriels invariants, c'est-à-dire, des espaces vectoriels qui restent inchangés sous l'application de ces isométries. De façon similaire, les isométries affines dans l'espace Affine Euclidien sont caractérisées par des points fixes associés. Le sous-espace vectoriel invariant de l'isométrie linéaire associée caractérise la nature des points fixes d'une isométrie. Donc, pour trouver les points fixes d'une isométrie, il suffit d'analyser l'isométrie linéaire associée.

Proposition 7 Soit une application affine $f : E \rightarrow E$:

- Si f a un point fixe x , l'ensemble des points fixes de f est $E(1, f) = x + \vec{E}(1, \vec{f})$.
- $\vec{E}(1, \vec{f}) = \{o\}$ si et seulement si f a un unique point fixe (et réciproquement).

Pour illustrer les résultats de cette proposition, supposons maintenant que f admette au moins un point fixe $x \in E$ et $p_+ = \dim(\vec{E}(1, \vec{f}))$. Alors, l'ensemble des points fixes de f est défini par un des cas suivants :

$p_+ = 0$	f a un unique point fixe $\{x\}$.
$p_+ = 1$	f a une droite D de points fixes passant par x et ayant pour direction du vecteur propre associé à la valeur propre 1.
$p_+ = 2$	f a un plan affine P de points fixes défini par les deux vecteurs propres associés à la valeur propre 1 et contenant x .
$p_+ = 3$	l'ensemble de points fixes de f est l'espace affine E .

TABLE 2.3 – Nature des points fixes, en supposant que f a au moins un point fixe.

Évaluation des points fixes

Pour trouver un point fixe $x \in E$ de f , il suffit de résoudre l'équation :

$$f(x) - x = 0.$$

Alors, fixons $O \in E$ comme origine, on déduit :

$$\overrightarrow{Of(x)} - \overrightarrow{Ox} = \overrightarrow{Of(O)} + \overrightarrow{f(O)f(x)} - \overrightarrow{Ox} = \overrightarrow{Of(O)} + \vec{f}(\overrightarrow{Ox}) - \overrightarrow{Ox} = 0.$$

Trouver le point fixe x revient donc à trouver \overrightarrow{Ox} tel que :

$$(\vec{f} - Id_{\vec{E}})\overrightarrow{Ox} = \overrightarrow{f(O)O}. \quad (2.15)$$

C'est-à-dire, le point fixe x est la solution du système linéaire (2.15).

Si $\vec{E}(1, \vec{f}) = \{o\}$, alors $\vec{f} - Id_{\vec{E}}$ est injective par hypothèse, donc bijective car $\dim(\vec{E})$ est finie, et il existe un unique $x \in E$ satisfaisant le système linéaire (2.15). En revanche, x est une des solutions de ce système linéaire.

Ici, nous constatons que, avec le support de la décomposition canonique des isométries, l'identification des isométries canoniques associées se base sur l'analyse des points fixes. Alors, nous définissons les trois isométries canoniques en fonction de la nature de leurs points fixes. De plus, puisque les coordonnées homogènes sont prises en compte, chaque isométrie canonique f est représentée sous forme matricielle telle que :

$$f : E \mapsto E, f(x) = Ax. \quad (2.16)$$

avec $A = [a_{ij}]_{4 \times 4}$, $a_{ij} \in \mathbb{R}$ et $x \in E$.

Définition 11 Translation

Une isométrie f est dite une **translation** si elle déplace tous les points dans E vers une direction fixée par une distance constante.

$$A_{Trans} = \begin{bmatrix} 1 & 0 & 0 & a_x \\ 0 & 1 & 0 & a_y \\ 0 & 0 & 1 & a_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.17)$$

où $\vec{a} = (a_x, a_y, a_z)$ est le vecteur de translation. Une translation n'a donc aucun point fixe.

Définition 12 Rotation

Une isométrie f est dite une **rotation** d'axe D si elle transforme les points de l'axe D en eux-mêmes et fait tourner les autres points x en x' d'un angle θ autour de cet axe.

$$A_{Rot} = \begin{bmatrix} d_x^2 + (1 - d_x^2)c & d_x d_y(1 - c) - d_z s & d_x d_z(1 - c) + d_y s & p_x \\ d_x d_y(1 - c) - d_z s & d_y^2 + (1 - d_y^2)c & d_y d_z(1 - c) + d_x s & p_y \\ d_x d_z(1 - c) + d_y s & d_y d_z(1 - c) + d_x s & d_z^2 + (1 - d_z^2)c & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.18)$$

où $c = \cos\theta$, $s = \sin\theta$, l'axe D a pour orientation le vecteur normalisé $\vec{D} = (d_x, d_y, d_z)$ et passe par le point $p = (p_x, p_y, p_z)$. Une rotation admet donc une droite D de points fixes.

Définition 13 Réflexion

Une isométrie f est dite une **réflexion** par rapport à un plan Δ si elle transforme tous les points du plan Δ en eux-mêmes et les autres points x en x' de l'autre côté de Δ tels que $\text{dist}(x, \Delta) = \text{dist}(x', \Delta)$, avec $\text{dist}(\cdot, \Delta)$ est la distance euclidienne orthogonale d'un point au plan Δ .

$$A_{Ref} = \begin{bmatrix} 1 - 2n_x^2 & -2n_x n_y & -2n_x n_z & p_x \\ -2n_x n_y & 1 - 2n_y^2 & -2n_y n_z & p_y \\ -2n_x n_z & -2n_y n_z & 1 - 2n_z^2 & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.19)$$

où le plan Δ est défini par la normale $\vec{n} = (n_x, n_y, n_z)$ et un point $p = (p_x, p_y, p_z)$. Une réflexion a donc un plan Δ de points fixes.

Grâce à la décomposition canonique et à la nature des points fixes, les isométries de l'espace à trois dimensions sont classées en différents groupes. Supposons une isométrie $f : E \rightarrow E$ associée à l'application linéaire $\vec{f} : \vec{E} \rightarrow \vec{E}$ et $f = g \circ t_{\vec{a}}$. Posons les notations suivantes :

Notation 1

- $p_+ = \dim \vec{E}(1, \vec{f})$ (respectivement $p_- = \dim \vec{E}(-1, \vec{f})$) : la dimension de l'espace propre associé à la valeur propre 1 (respectivement -1) de \vec{f} . p_+ est impair (1 ou 3) si et seulement si f est directe ; p_+ est pair (0 ou 2) si et seulement si f est indirecte. De plus, $p_+ + p_- \leq 3$.
- $\Lambda = \{x \in X, f(x) = x\}$: l'ensemble des points fixes de f .
- $\Omega = \{x \in X, g(x) = x\}$: l'ensemble des points fixes de g .
- D : la droite des points fixes de g si $p_+ = 1$.
- P : le plan des points fixes de g si $p_+ = 0$ ou $p_+ = 2$.
- x : un point fixe de f s'il existe.
- $t_{\vec{a}}$: la translation du vecteur \vec{a} .
- $r(D, \theta)$: la rotation autour d'axe D d'un angle θ .
- $s(P)$: la symétrie orthogonale par rapport au plan P .

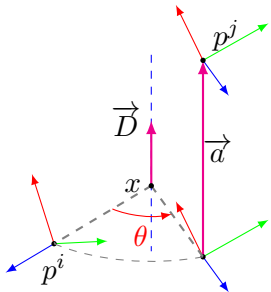
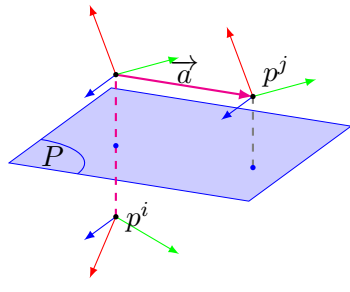
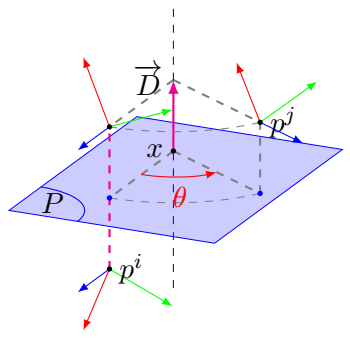
Alors, les classes des isométries différentes de l'identité sont les suivantes :

Classe	p_+, p_-	Λ, Ω	\vec{a}	f	Description	Remarques
TRAN	$p_+ = 3$ $p_- = 0$	$\Lambda = \emptyset$ $\Omega = E$	$\vec{a} \neq 0$	$t_{\vec{a}}$	f est une translation.	
ROT	$p_+ = 1$ $p_- = 0$ ou $p_- = 2$	$\Lambda = \Omega$ $= D$	$\vec{a} = 0$	$r(D, \theta)$	f est une rotation autour de l'axe D et d'angle $\theta \in [-\pi, \pi]$.	
TROT	$p_+ = 1$ $p_- = 0$ ou $p_- = 2$	$\Lambda = \emptyset$ $\Omega = D$	$\vec{a} \neq 0$	$r(D, \theta) \circ t_{\vec{a}}$	f est composée d'une rotation autour de l'axe D et d'angle $\theta \in [-\pi, \pi]$ et d'une translation $t_{\vec{a}}$. f est un "vissage".	$\vec{a} \parallel D$
SYM	$p_+ = 2$ $p_- = 1$	$\Lambda = \Omega$ $= P$	$\vec{a} = 0$	$s(P)$	f est une symétrie orthogonale par rapport au plan P .	
VIS	$p_+ = 2$ $p_- = 1$	$\Lambda = \emptyset$ $\Omega = P$	$\vec{a} \neq 0$	$s(P) \circ t_{\vec{a}}$	f est composée d'une symétrie orthogonale par rapport au plan P et d'une translation $t_{\vec{a}}$.	$\vec{a} \parallel P$

SROT	$p_+ = 0$ $p_- = 1$	$\Lambda = \Omega$ $= \{x\}$	$\vec{a} = 0$	$s(P)$ $r(D, \theta)$	\circ	f est composée d'une symétrie orthogonale par rapport au plan P et d'une rotation d'axe D (perpendiculaire au plan P) d'angle $\theta \in [-\pi, \pi]$.	Le vecteur propre associé à la valeur propre -1 de f est d'une part la normale du plan P et d'autre part la direction de D .
------	------------------------	---------------------------------	---------------	--------------------------	---------	---	--

TABLE 2.4 – Liste des classes d'isométries.

Le tableau 2.5 illustre en images les différentes classes d'isométries autres que l'identité en fonction du type d'isométrie et de la nature des points fixes.

Isométrie	Directe	Indirecte
Sans point fixe	(Classe TROT) $f = r(D, \theta) \circ t_{\vec{a}}$ 	(Classe VIS) $f = s(P) \circ t_{\vec{a}}$ 
Un point fixe	\emptyset	(Classe SROT) $f = s(P) \circ r(D, \theta)$ 
Ligne de points fixes	(Classe ROT) $f = r(D, \theta)$	\emptyset

Plan de points fixes	\emptyset	(Classe SYM) $f = s(P)$
Tous les points sont fixes	(Classe TRAN) $f = t_{\vec{a}}$	\emptyset

TABLE 2.5 – Illustrations des classes d'isométries.

Dans ce paragraphe, nous avons révisé la notion de l'espace affine euclidien, un espace métrique dont la distance euclidienne est une mesure. Ainsi, nous avons défini les différents types d'isométries en fonction des points fixes, qui sont aussi appelés **points invariants** ou **invariants** [Cederberg 2001, Johnson 2001, Fresnel 2010]. Maintenant, nous avons tous les moyens nécessaires pour définir et analyser la notion principale de ce travail, l'**isométrie partielle**, qui est détaillée dans le paragraphe suivant.

3.2 Groupe des isométries partielles

Définition 14 Isométrie partielle

Soient M un modèle 3D dans l'espace affine euclidien E , $M_1, M_2 \subset M$ deux parties connexes différentes de ce modèle ($\overset{\circ}{M}_1 \cap \overset{\circ}{M}_2 = \emptyset$). Alors, il existe une **isométrie partielle** entre ces deux patchs si et seulement s'il y a une isométrie $f \in \mathcal{O}(E)$ telle que :

$$f(M_1) = M_2. \quad (2.20)$$

Dans ce cas, M_1 et M_2 sont dites similaires à une isométrie près.

3.2.1 Classification des isométries

Nous voulons définir les caractéristiques des isométries en décomposant ces isométries en transformations canoniques. L'analyse globale d'une isométrie f se divise en trois étapes :

1. Décomposition canonique de f en une translation de vecteur \vec{a} et une isométrie ayant au moins un point fixe g telles que $f = g \circ t_{\vec{a}} = t_{\vec{a}} \circ g$.
2. Classification de f en fonction de ses points fixes.
3. Caractérisation de f par un minimum de paramètres.

En décomposant une isométrie de façon canonique et en analysant la nature de ses points fixes, l'isométrie est considérée comme une composition des transformations canoniques : translation, rotation autour d'un axe et la réflexion. Cette décomposition fournit des invariants naturels et des métriques associées peuvent être définies. En effet :

- La translation est caractérisée par un vecteur $\vec{a} \in \mathbb{R}^3$.
- La rotation est caractérisée par un angle $\theta \in \mathbb{R}$ et un axe D (ayant pour direction $\vec{n} \in \mathbb{R}^3$ ($\|\vec{n}\| = 1$) et passant par le point fixe $x \in \mathbb{R}^3$).
- La réflexion est caractérisée par un plan P ayant pour normale $\vec{n} \in \mathbb{R}^3$ ($\|\vec{n}\| = 1$) et contenant un point fixe $x \in \mathbb{R}^3$.
- La symétrie centrale est caractérisée par un point fixe $x \in \mathbb{R}^3$.

Alors, les invariants des classes d'isométries autres que l'identité peuvent être déterminés par le tableau suivant :

Isométrie	Classe	f	\vec{a}	\vec{n}	x	θ
Directe	TRAN	$t_{\vec{a}}$	✓			
	ROT	$r(D, \theta)$		✓	✓	✓
	TROT	$r(D, \theta) \circ t_{\vec{a}}$	✓	✓	✓	✓
Indirecte	SYM	$s(P)$		✓	✓	
	VIS	$s(P) \circ t_{\vec{a}}$	✓	✓	✓	
	SROT	$s(P) \circ r(D, \theta)$		✓	✓	✓

TABLE 2.6 – Invariants des classes d'isométries.

3.2.2 Métriques des isométries

La décomposition des isométries en transformations canoniques est avantageuse pour le calcul des métriques d'isométries, particulièrement pour la comparaison des isométries.

En dérivant les résultats analysés dans le tableau 2.6, supposons deux isométries

f et f' d'une même classe, nous visons à estimer la distance entre elles, notée $d(f, f')$.

Posons aussi :

- $dist(.,.)$ la distance euclidienne entre deux entités qui sont une combinaison d'un ensemble de points, de droites ou de plans.
- B la largeur de diagonale de la boîte englobante du modèle définie par l'équation (2.13) qui correspond à la normalisation proposée par [Mitra 2006] pour pondérer entre les rotations et les translations.

Pour dériver la distance entre deux isométries f et f' d'une même classe, nous exploitons les propriétés des transformations canoniques.

- Deux **translations** $f = t_{\vec{a}}$ et $f' = t_{\vec{a}'}$ (isométries de la classe TRAN) sont identiques si elles sont de même direction et de même longueur. Supposons que \vec{a} et \vec{a}' sont respectivement des vecteurs de translation de f et f' , on a :

$$d(f, f') = \left(1 - \frac{\langle \vec{a}, \vec{a}' \rangle}{\|\vec{a}\| \|\vec{a}'\|}\right) + \frac{\|\vec{a}\| - \|\vec{a}'\|}{\max(\|\vec{a}\|, \|\vec{a}'\|)}$$

Le premier terme mesure l'écart entre les directions normalisées \vec{a} et \vec{a}' , le deuxième terme mesure l'écart à la longueur de ces directions.

- Deux **rotations** $f = r(D, \theta)$ et $f' = r(D', \theta')$ (isométries de la classe ROT) sont identiques si leurs axes de rotation coïncident et leurs angles de rotation sont les mêmes. Supposons que f est une rotation autour de l'axe D d'un angle θ , f' une rotation autour de l'axe D' d'un angle θ' et que \vec{n} et \vec{n}' sont des directions des axes D et D' , on a :

$$d(f, f') = \left(1 - \langle \vec{n}, \vec{n}' \rangle\right) + \frac{dist(D, D')}{B} + \left(\frac{|\theta - \theta'|}{2\pi}\right)$$

Les deux premiers termes représentent l'écart aux directions et aux positions entre deux axes D et D' . Le troisième terme mesure l'écart angulaire de ces rotations.

- Deux **symétries** $f = s(P)$ et $f' = s(P')$ (isométries de la classe SYM) sont les mêmes si leurs plans de symétrie P et P' coïncident. Il suffit donc de considérer la direction des normales \vec{n} et \vec{n}' et la distance entre ces plans. Pourtant, le calcul de la distance entre les plans n'est pas robuste. En effet, s'il y a une perturbation légère sur la direction des normales de deux plans coïncidents, les deux plans s'intersectent et la distance entre eux est nulle. Nous proposons donc d'utiliser en plus les deux paires de points (p_i, p_j) et (p'_i, p'_j) qui associent à f et f' . Soient m et m' les milieux de (p_i, p_j) et (p'_i, p'_j) respectivement. Alors :

$$d(f, f') = \left(1 - \langle \vec{n}, \vec{n}' \rangle\right) + \frac{dist(m, P')}{B} + \frac{dist(m', P)}{B}$$

Le premier terme mesure l'écart entre les normales. Les deux derniers termes mesurent l'écart entre les deux plans de symétrie respectifs.

Les isométries des trois classes précédentes se composent seulement une transformation canonique. Par contre, les isométries des classes restantes se composent de deux transformations canoniques. Alors, la comparaison de deux isométries d'une même classe peut s'effectuer en combinant les comparaisons des transformations canoniques existantes dans cette classe. Néanmoins, il y a des composantes qui se répètent dans certaines classes. On peut enlever une des composantes dans chaque contexte.

- Chaque isométrie de la classe TROT se compose d'une rotation et d'une translation, l'axe de rotation et la translation ont la même direction. La distance entre deux isométries $f = r(D, \theta) \circ t_{\vec{a}}$ et $f' = r(D', \theta') \circ t_{\vec{a'}}$ est :

$$d(f, f') = \left(1 - \frac{|\langle \vec{a}, \vec{a'} \rangle|}{\|\vec{a}\| \|\vec{a'}\|}\right) + \frac{\|\vec{a}\| - \|\vec{a'}\|}{\max(\|\vec{a}\|, \|\vec{a'}\|)} + \frac{\text{dist}(D, D')}{B} + \left(\frac{|\theta - \theta'|}{2\pi}\right)$$

Le premier terme représente l'écart aux directions entre les translations ainsi qu'entre les axes de rotation. Le deuxième terme mesure l'écart à la longueur des deux vecteurs de translation. Le troisième terme représente l'écart à la position entre les axes de rotation et le quatrième représente l'écart angulaire des rotations.

- Chaque isométrie de la classe SROT se compose d'une rotation et d'une réflexion, l'axe de rotation et la normale du plan de symétrie ont la même direction. La distance entre deux isométries $f = s(P) \circ r(D, \theta)$ et $f' = s(P') \circ r(D', \theta')$ est :

$$d(f, f') = \left(1 - |\langle \vec{n}, \vec{n'} \rangle|\right) + \frac{\text{dist}(m, P')}{B} + \frac{\text{dist}(m', P)}{B} + \frac{\text{dist}(D, D')}{B} + \left(\frac{|\theta - \theta'|}{2\pi}\right)$$

Le premier terme représente l'écart à la direction entre les normales de plans de symétrie ainsi qu'entre les axes de rotation. Le deuxième et le troisième termes représentent l'écart entre les plans de symétrie. Le quatrième terme mesure l'écart à la position des axes de rotation et le dernier représente l'écart angulaire des rotations.

- Chaque isométrie de la classe VIS se compose d'une symétrie et d'une translation, la normale du plan de symétrie et la direction de translation sont perpendiculaires. La distance entre deux isométries $f = s(P) \circ t_{\vec{a}}$ et $f' = s(P') \circ t_{\vec{a'}}$ est :

$$d(f, f') = \left(1 - |\langle \vec{n}, \vec{n'} \rangle|\right) + \frac{\text{dist}(m, P')}{B} + \frac{\text{dist}(m', P)}{B} + \frac{\|\vec{a}\| - \|\vec{a'}\|}{\max(\|\vec{a}\|, \|\vec{a'}\|)}$$

Le premier terme mesure l'écart à la direction entre les normales ainsi qu'entre les translations. Le deuxième et troisième termes représentent l'écart entre

les plans de symétrie. Le quatrième terme mesure l'écart à la longueur entre les translations.

4 Conclusion

Dans ce chapitre, nous avons fait un tour d'horizon du domaine de la *Modélisation Géométrique* et des méthodes de modélisation utilisées dans le monde 3D. Nous avons aussi résumé les avantages et les inconvénients de ces méthodes. A partir de là, nous avons choisi les modèles BRep-NURBS, un standard dans le domaine de la CAO, comme base de notre recherche.

Ensuite, nous avons présenté nos algorithmes d'échantillonnage des faces du modèle BRep-NURBS. Ces algorithmes sont appropriés pour notre algorithme de détection de similarités : les échantillons sont générés à l'intérieur des faces et sont aussi espacés de façon relativement uniforme, la grille régulière des échantillons permet la navigation aux voisins d'un échantillon donné. Puis, nous avons introduit les isométries dans l'espace affine euclidien, un facteur important contribuant au succès de la détection de similarités dans ce travail. En effet, les isométries sont considérées comme des transformations canoniques : rotations, translations, réflexions ou leurs compositions. La décomposition et la classification des isométries en transformations canoniques sont réalisées par le biais des points fixes qui, à leur tour, fournissent des caractéristiques invariantes des isométries.

Dans le chapitre suivant, nous allons introduire notre algorithme de détection de similarités. Étant donné un modèle BRep-NURBS, cet algorithme peut identifier non seulement les régions locales similaires mais aussi les transformations canoniques qui les lient.

Détection d'isométries partielles

Sommaire

1	Travaux précédents	44
1.1	Procédé de type transformée de Hough	46
1.2	Détection de similarités dans des modèles BRep	49
1.3	Conclusion	51
2	Algorithme de détection	52
2.1	Utilisation de la transformée de Hough	52
2.2	Algorithme proposé	54
3	Extraction des isométries	54
3.1	Extraction basée topologie	55
3.2	Extraction basée géométrie	60
3.3	Caractérisation des isométries	65
3.4	Conclusion	66
4	Classification des isométries	66
4.1	Méthode Mean Shift	66
4.2	Méthode Spectrale	69
4.3	Application de la classification spectrale pour la recherche de similarités	73
5	Validation	73
5.1	Validation itérative	74
5.2	Validation par pondération	74
5.3	Validation basée histogramme	75
5.4	Validation basée distance de Hausdorff	76
6	Extension	77
7	Expérimentations	77
8	Conclusion	81

Dans ce chapitre, nous présentons des algorithmes qui permettent d'une part d'identifier automatiquement les différentes parties répétées dans un modèle paramétrique M défini dans l'espace affine euclidien E , et d'autre part d'estimer la transformation entre ces parties. Effectivement, les tuples de la forme $(\{(M_1, M_2)\}, I)$ sont à déterminer, où (M_1, M_2) sont des couples de parties similaires à l'isométrie près I et $M_1, M_2 \subset M, M_1 \cap M_2 \neq \emptyset$, tels que :

$$I(M_1) = M_2. \quad (3.1)$$

Dans ce travail, les isométries considérées sont des composées de rotations, translations et réflexions. Supposons une telle isométrie définie par l'équation :

$$I(P) = R * P + t. \quad (3.2)$$

où $P \in E$, $R \in \mathcal{O}(\vec{E})$ la matrice de rotation et $t \in \mathbb{R}^3$ le vecteur de translation.

1 Travaux précédents

La **détection de similarités** est un sujet étudié largement dans le domaine de l'image numérique en 2D mais aussi en modélisation 3D. En général, les algorithmes existants se distinguent par le choix des caractéristiques à évaluer dans des modèles ou des surfaces 3D. Ces caractéristiques sont adaptées à la nature du modèle et identifient différentes transformations. Les algorithmes de détection de similarités sont donc classifiés en catégories suivant les méthodes utilisées, les niveaux de similarités concernés, les modèles à l'entrée et les types de résultats. Les travaux de Hel-Or [Hel-Or 2010] et de Mitra [Mitra 2012] ont classifié et détaillé les travaux concernant la détection de similarités. Il existe trois paradigmes de détection des similarités :

- Similarités intrinsèques vs. extrinsèques.
- Similarités exactes vs. approximatives.
- Similarités globales vs. locales.

Par définition [Ovsjanikov 2008], les similarités extrinsèques d'une forme 3D sont invariantes sous les transformations rigides. De même, les similarités intrinsèques sont invariantes sous les autres transformations non-rigides comme les transformations de Möbius, les opérateurs Laplace Beltrami [Mitra 2012] (figure 3.1). Comme nous l'avons mentionné dans le chapitre précédent, nous utilisons les isométries pour identifier des types de similarités existantes dans des modèles BRep-NURBS. Nous nous intéressons seulement à la détection des similarités extrinsèques. De plus, un travail précurseur d'Atallah et al. [Atallah 1985] utilise la correspondance des motifs pour identifier les similarités exactes entre des ensembles de points 2D ou 3D. Néanmoins, comme il existe des tolérances numériques lors la production ou l'acquisition des modèles 3D, la détection des similarités approximatives a été plus étudiée. Dans ce paragraphe, nous résumons les algorithmes récents correspondants à différentes approches indépendamment du type de modèle 3D en entrée.



FIGURE 3.1 – Objets exposant la symétrie planaire extrinsèque (à gauche) et intrinsèque (à droite) [Mitra 2012].

Sun et al. [Sun 1997] convertissent le problème de détection de symétries en calcul de corrélation de l'image Gaussienne. Ils ont observé que si un objet est symétrique, son image Gaussienne l'est aussi (figure 3.2). Alors, en fonction de la rotation ou de la réflexion, l'image Gaussienne est estimée à partir des échantillons et est convertie en histogramme d'orientation du modèle. Cet histogramme est ensuite considéré pour identifier la rotation et la réflexion dans le modèle.

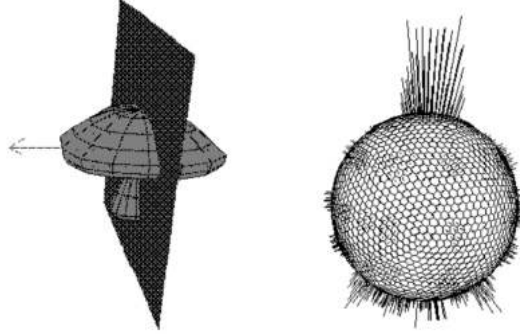


FIGURE 3.2 – Réflexion d'un modèle de champignon et son histogramme d'orientation correspondante [Sun 1997].

Zabrodsky et al. [Zabrodsky 1995] ont proposé la distance symétrique comme une métrique mesurant la symétrie locale présente dans des objets 3D. Cette distance est définie par la moyenne des carrés de la distance nécessaire pour déplacer les points de l'objet original aux points correspondants de l'objet symétrique. La figure 3.3 montre le processus du calcul de cette distance. La forme symétrique $\hat{P} = \{\hat{P}_i\}_{i=0}^{n-1}$ (image c) est estimée par une *fonction de symétrie* à partir de la forme originale $P = \{P_i\}_{i=0}^{n-1}$ (image a). La distance de symétrie SD est définie par l'équation :

$$SD = \frac{1}{n} \sum_{i=0}^{n-1} \|P_i - \hat{P}_i\|^2 \quad (3.3)$$

Alors, la distance SD est comprise sur un intervalle réel déterminant le niveau de symétrie d'une forme, SD est nulle pour les formes parfaitement symétriques. En utilisant cette distance, les régions locales symétriques d'un objet 3D peuvent être identifiées. Pourtant, cet algorithme considère seulement les mesures continues des similarités et ne peut pas vérifier s'il existe une similarité ou une anti-similarité [Mills 2001] car il existe un seuil de comparaison à régler. De plus, cet algorithme ne peut pas estimer la transformation entre deux formes similaires. Il n'est donc pas approprié pour nos objectifs dans ce travail.

Kazhdan et al. [Kazhdan 2003] introduisent un descripteur de réflexion qui représente les symétries d'un objet 3D par rapport à tous les plans possibles passant par le barycentre de l'objet. Ce descripteur est estimé par une fonction sphérique mesurant l'invariance de la réflexion du modèle 3D par rapport aux plans passant par le barycentre (figure 3.4). Chaouch et al. [Chaouch 2008] adaptent ce descripteur de symétrie pour identifier la symétrie planaire dominante d'un modèle 3D. Contrairement au descripteur original, leur descripteur représente les symétries par rapport

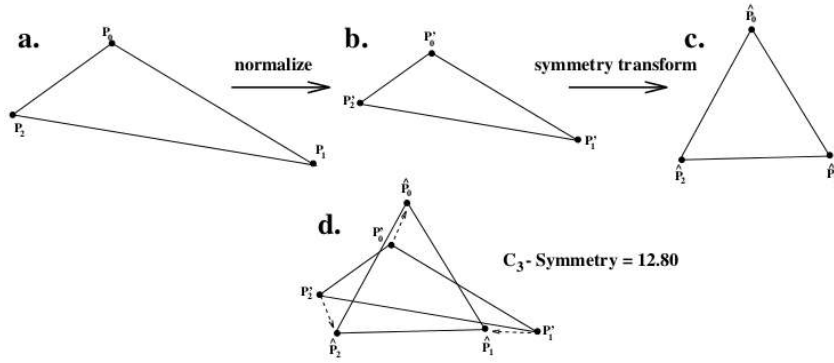


FIGURE 3.3 – Calcul de la distance de symétrie [Zabrodsky 1995].

seulement aux plans passant par le centre de gravité du modèle. Le centre de gravité et les plans initiaux de l'algorithme sont estimés par l'*Analyse des Composantes Principales Continues* (CPCA). L'algorithme teste tous les plans symétriques qui tournent avec angle donné autour des plans initiaux pour choisir le meilleur. Malgré la modification proposée à CPCA pour résoudre le problème de sensibilité à la distribution de points, cette proposition n'est pas applicable à toutes formes 3D [Papadakis 2007].

Une approche récente, appelée *votes dans l'espace de transformation* ou *procédé du style de Hough*, est utilisée dans plusieurs travaux détectant des similarités partielles [Loy 2006, Podolak 2006, Mitra 2006] et est détaillée dans le paragraphe suivant.

1.1 Procédé de type transformée de Hough

Cette approche partage le même principe de traitement que la *transformée de Hough* en imagerie. Son idée principale est basée sur une observation où les correspondances entre les parties similaires d'un modèle partagent les mêmes caractéristiques ou invariants : « À partir d'un ensemble de points d'échantillons de la surface d'un modèle, les caractéristiques de ces points sont estimées. Les points de mêmes caractéristiques sont mis en correspondance pour former un vote quantifiant

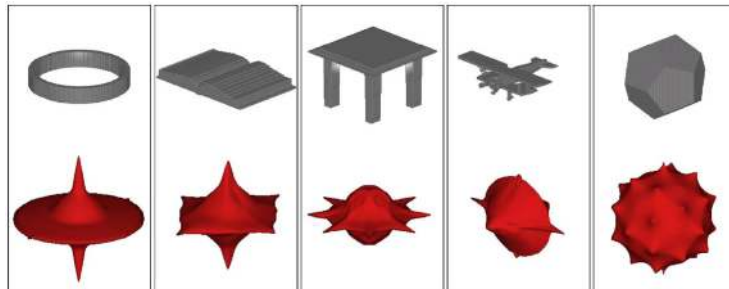


FIGURE 3.4 – Les descripteurs de réflexion de quelques modèles [Kazhdan 2003].

la transformation entre eux. Ces votes sont ensuite accumulés dans un espace de transformation afin de déterminer les groupes dominants qui représentent potentiellement des couples de régions similaires » [Mitra 2006].

Les travaux se différencient par l'utilisation des caractéristiques dans les modèles en entrée. Loy et al. [Loy 2006] utilisent le descripteur SIFT (Scale-Invariant Feature Transform en anglais) extraits des points d'une image en 2D, chaque point est caractérisé par un vecteur composé de sa position, de son orientation et de son échelle. Les orientations des points sont ensuite normalisées par rapport à une référence. En comparant avec les orientations de ces points, les paires de points, où il existe peut-être une réflexion ou une rotation entre eux, sont formés. Les lignes de réflexion et les centres de rotations sont fournis à l'espace de transformation pour accumuler les similarités dominantes.

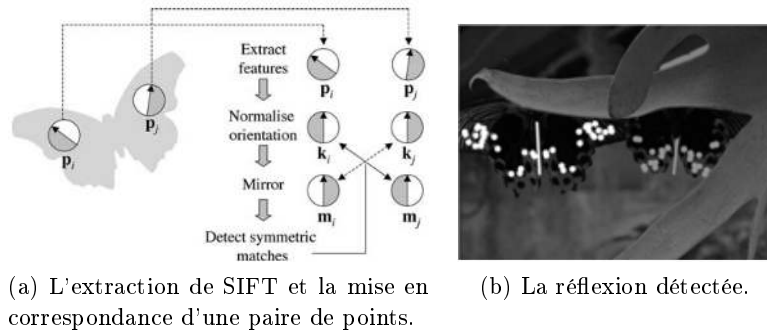


FIGURE 3.5 – Descripteur SIFT pour détecter la symétrie dans une image [Loy 2006].

Podolak et al. [Podolak 2006] introduit la *transformation de réflexion planaire* (*PRST*, Planar-Reflective Symmetry Transform en anglais) partageant le même principe de distance de symétrie [Zabrodsky 1995]. Cette transformation définie à partir de l'espace des points et de l'ensemble des plans, fournit une mesure continue de la réflexion de l'objet par rapport à tous les plans qui intersectent sa boîte englobante. La réflexion finale de l'objet est identifiée par le maximum local des mesures continues fournies par la *PRST*. En d'autres termes, des plans de réflexions entre des paires de points sont considérés pour accumuler les plans de réflexions dominants dans des images 2D ou des modèles 3D (figure 3.6).

Mitra et al. [Mitra 2006] définissent le descripteur local d'un point comme une composition des courbures principales et des directions principales évaluées en ce point. Ainsi, les directions principales sont utilisées pour définir le repère local du point. Deux points sont mis en correspondance si et seulement s'ils ont les mêmes courbures. La transformation entre ces deux points est estimée telle que le repère local d'un point coïncide avec celui de l'autre point (figure 3.7a). Les angles d'Euler, la translation et un facteur d'échelle correspondant à la transformation estimée caractérisent la paire de points dans l'espace des transformations où les similarités dominantes sont identifiées. La figure 3.7b illustre l'accumulation des similarités dominantes dans l'espace de transformations. Ici, dans le modèle 3D du château de Chambord, les deux plus grands modes représentent les deux réflexions dominantes

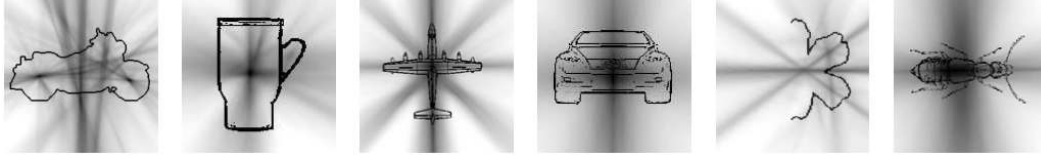
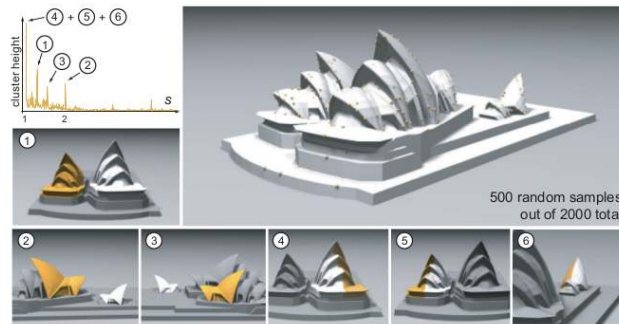
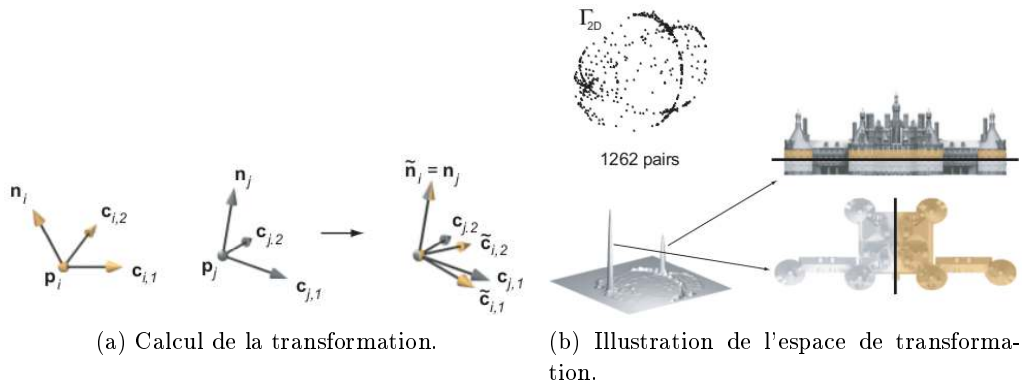


FIGURE 3.6 – La PRST capture le niveau de la réflexion des objets par rapport à tous les plans dans l'espace. Les lignes plus sombres représentent les réflexions plus importantes.

du modèle.

La figure 3.7c illustre quelques résultats du travail de [Mitra 2006]. L'algorithme proposé peut identifier tous types de similarités partielles d'un modèle 3D. Par exemple, (1), (2), (3) présentent la mise en échelle suivie d'une rotation, (4), (5), (6) présentent la symétrie entre les parties. Cet algorithme satisfait nos objectifs dans la détection de similarités et est le point de départ de notre travail.

Nous constatons que tous les modèles en entrée des algorithmes précédents sont des maillages 3D. Dans la partie suivante, nous révisons quelques travaux sur la détection des similarités des surfaces paramétriques et des modèles BRep.

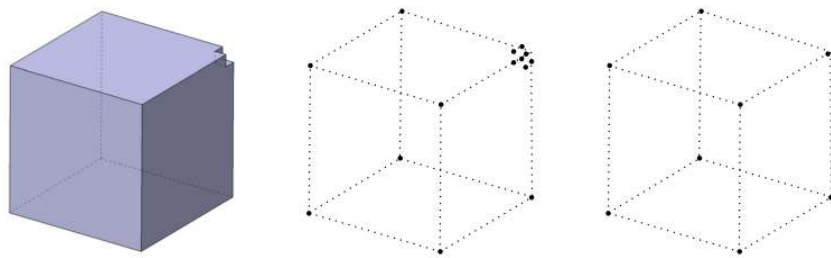


(c) Résultats de la détection.

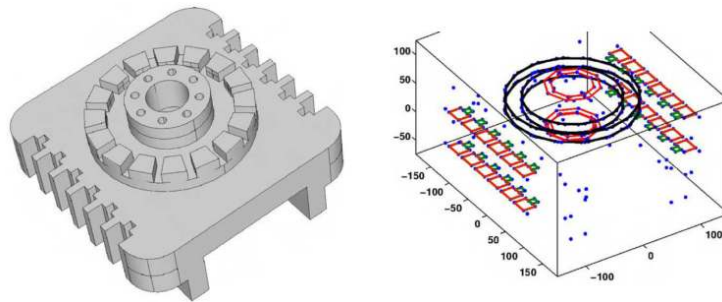
FIGURE 3.7 – La détection de similarités dans le travail de Mitra [Mitra 2006].

1.2 Détection de similarités dans des modèles BRep

En général, pour la détection de similarités, les modèles BRep sont considérés soit comme des modèles de points discrets soit comme des modèles continus. Dans la première approche, les arêtes des modèles BRep sont caractérisés par des ensembles de points [Mills 2001, Li 2008] (figure 3.8). De là, les traitements spécifiques sont effectués pour identifier les symétries existantes dans des modèles. Cependant, en ignorant les informations continues des courbes et des surfaces du modèle, la détection de symétrie dans ces modèles discrétisés est biaisée [Li 2011]. Les plans de symétries sont alors estimés de façon approximative avec une certaine tolérance.



(a) [Mills 2001]



(b) [Li 2008]

FIGURE 3.8 – La traduction des modèles BRep en modèles de point discrets.

Dans une recherche [Tate 2003], Tate et al. introduisent un algorithme identifiant les similarités partielles dans les modèles BRep. Leur algorithme utilise la comparaison des propriétés des *loops* extérieures des faces pour identifier les axes et les plans de symétrie. Cet algorithme est décomposé en cinq étapes :

- Estimer les propriétés des *loops* extérieures.
- Mettre en correspondance des *loops*.
- Construire les axes et les plans de symétrie.
- Justifier les axes et les plans de symétrie.
- Extraire les axes primaires de symétrie.

Les propriétés considérées des *loops* sont : l'aire, le centroïde et la normale. Les *loops* ayant la même aire sont mises en correspondance pour estimer les axes et les plans de symétrie. En général, un plan de symétrie est déterminé par un point p et une normale v_n . Étant données deux *loops* correspondantes, supposons que $c1$ et $c2$

soient ses centroïdes, v_1 et v_2 soient ses normales. Le plan de symétrie entre deux loops correspondantes est estimé comme illustré dans la figure 3.9.

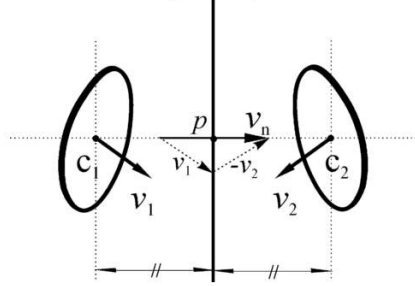


FIGURE 3.9 – Construction du plan de symétrie.

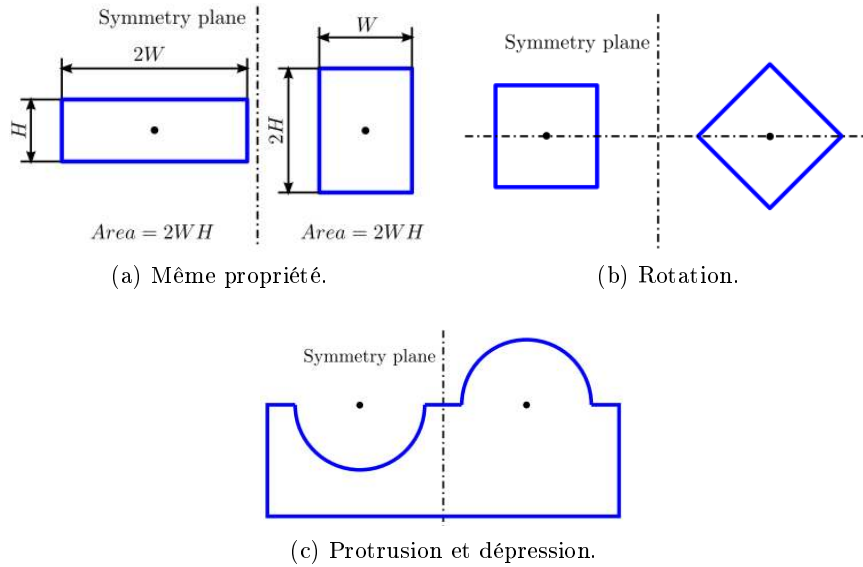


FIGURE 3.10 – Les limitations dans l'algorithme proposé dans [Tate 2003].

Malgré le succès de cet algorithme dans la détection de la symétrie partielle, il existe des faux positifs : les faces ayant les *loops* de mêmes propriétés sont identifiées comme symétriques alors qu'elles ne le sont pas. La figure 3.10 illustre un tel cas : l'information de l'objet réel est perdue lors de l'estimation des propriétés de ses *loops*.

Li propose d'utiliser aussi les arêtes pour identifier la symétrie dans des modèles BRep [Li 2011]. Des hypergraphes sont construits pour décrire les ares des modèles. Chaque hypergraphe décrit la relation d'adjacence des entités topologiques des bords comme les faces, les arêtes et les sommets. Il y a trois types d'hypergraphes : faces-arêtes, arêtes-sommets et faces-sommets (figure 3.11c). Ces hypergraphes permettent de générer les faces maximales regroupant toutes les faces adjacentes et les arêtes maximales regroupant les arêtes adjacentes. De plus, les surfaces sous-jacentes aux

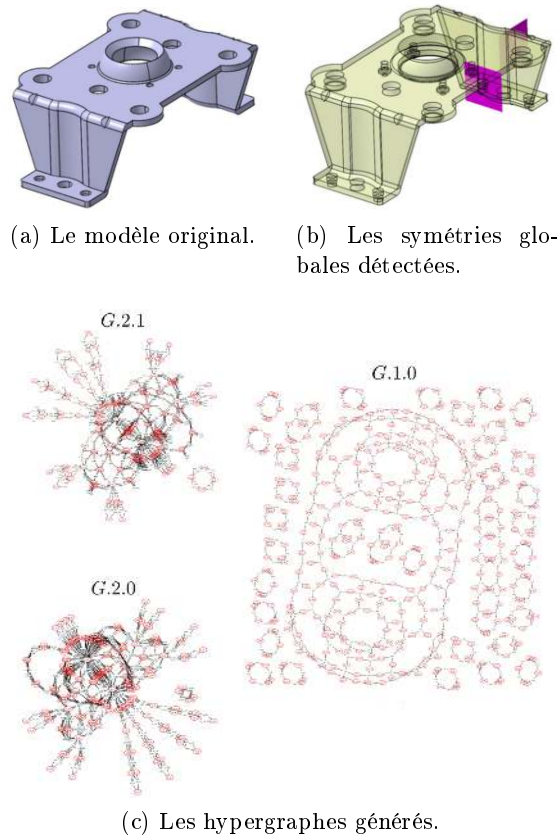


FIGURE 3.11 – Détection de symétrie dans des modèles BRep [Li 2011].

faces sont considérées comme des formes primaires telles que des sphères, des cubes, des cônes et des tores. Ces formes permettent de définir les contraintes pour identifier les plans de symétrie dans des faces. À partir des plans de réflexion candidats, une étape de propagation est effectuée en se basant sur les hypergraphes et sur les faces maximales pour trouver les réflexions planaires globales.

1.3 Conclusion

Nous avons révisé quelques algorithmes de détection de similarités dont les modèles en entrée sont soit des maillages soit des BReps. Chaque algorithme possède ses points forts ainsi que ses points faibles, il n'existe pas un algorithme commun ou un procédé standard pour tous types de modèles 3D. En général, alors que les algorithmes pour des maillages exploitent les propriétés de la géométrie algébrique et différentielle des points de la surface, les algorithmes pour les BReps exploitent de la topologie des BReps pour identifier les similarités. Les travaux de [Tate 2003, Li 2011] sont très proches de ce sujet de thèse, ce sont des bonnes références pour nous. Cependant, alors que nous nous intéressons aux modèles BRep-NURBS et ces travaux considèrent des modèles BRep dont les surfaces sous-jacentes sont limitées à des

primitives paramétriques telles que des sphères, des cubes, des cônes et des tores. De plus, ces travaux proposent des algorithmes dédiés à la détection des symétries planaires dans des modèles. Ils ne répondent pas bien à notre besoin de détecter tout type de similarités (sans compter la mise à l'échelle). Néanmoins, le *Procédé de la transformée de Hough* proposée dans le travail de Mitra [Mitra 2006] génère plusieurs recherches [Berner 2008, Bokeloh 2009]. Ce procédé est capable de capturer tous types de similarités entre deux régions des maillages. Dans ce travail, nous adaptons ce procédé pour détecter les similarités des modèles BReps basés NURBS. Nous appliquons en même temps à ce procédé les caractéristiques géométriques différentielles des surfaces NURBS et les caractéristiques topologiques des BReps pour répondre à nos besoins.

2 Algorithme de détection

Dans cette section, nous présentons le *Procédé de la transformée de Hough* pour la détection des similarités dans des modèles 3D, puis des définitions de similarité nécessaires pour dériver l'algorithme que nous proposons. Par la suite, nous notons $M = \{F^i\}, i \in [0, n_F)$, le modèle BRep-NURBS composé de n_F faces. Chaque face est définie par $F^i = \{S^i, V^i\}$, où S^i est la surface NURBS sous-jacente et $V^i = \{v_0^i, \dots, v_{n-1}^i\}$ est l'ensemble de n sommets qui limite F^i . Notre algorithme identifie automatiquement non seulement les faces similaires dans un modèle donné mais aussi estime les isométries entre elles. C'est-à-dire, si deux patches $P^1 = \{F^k\}$ et $P^2 = \{F^k\}$ sont similaires, pour $k \in [0, n_P)$ et $n_P < \frac{n_F}{2}$, nous identifions une isométrie T telle que :

$$P^2 = T(P^1). \quad (3.4)$$

Nous considérons les isométries comme des compositions d'isométries canoniques : les rotations, les translations et les réflexions.

2.1 Utilisation de la transformée de Hough

Dans la suite de cette section, nous introduisons un nouveau concept : les *entités de référence*. Ce sont des unités géométriques à partir desquelles nous pouvons évaluer les caractéristiques géométriques ou topologiques et estimer la transformation affine de l'autre entité. Dans ce travail, nous utilisons deux types d'*entités de référence* : les points de la surface NURBS et les faces de l'objet BRep.

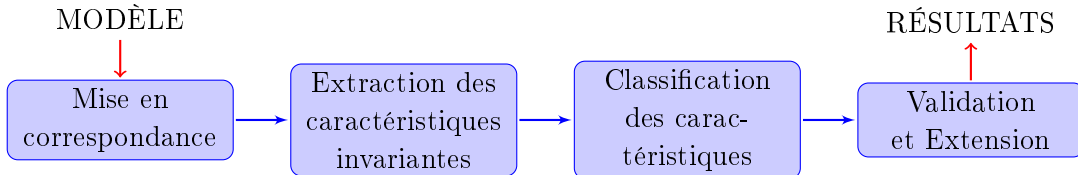


FIGURE 3.12 – Procédé par vote.

En général, après avoir défini les *entités de référence* et les *caractéristiques invariantes* des transformations, cette approche se compose de plusieurs phases consécutives comme illustré dans la figure 3.12 :

1. *Mise en correspondance* : des entités de référence sont prélevées du modèle suivant une technique appropriée, elles sont caractérisées par des signatures. Ensuite, les entités de même signature sont mises en correspondance et forment des *paires*.
2. *Extraction des caractéristiques invariantes* : pour chaque paire, une transformation entre les deux entités est évaluée pour extraire des caractéristiques invariantes.
3. *Classification des caractéristiques* : les paires de mêmes caractéristiques sont groupées. Chaque groupe représente potentiellement deux parties similaires du modèle.
4. *Validation et Extension* : dans la plupart des cas, les paires dans chaque groupe ne forment pas des parties entières similaires. De plus, la connectivité et la cohérence des paires ne sont pas assurées. Cette étape est donc nécessaire.

Dans le contexte de notre travail, un objet BRep est constitué de plusieurs *faces* et un modèle BRep se compose de nombreux objets BRep. Le modèle de la figure 3.13 représentant un violon est composé de plusieurs objets BRep de couleurs différentes (figure 3.13a). De même, l'objet BRep représentant la table d'harmonie et le manche (en vert) contient de nombreuses faces (figure 3.13b). Une *face* (figure 3.13c) est une surface NURBS (en violet) limitée par des *arêtes* (en vert) et des *sommets* (en rouge). En d'autres termes, un modèle BRep contient de nombreuses *faces*. Le problème de la détection des similarités d'un modèle se réduit à la détection des similarités entre les *faces* de ce modèle. Pour une *face*, la *loop* extérieure trace

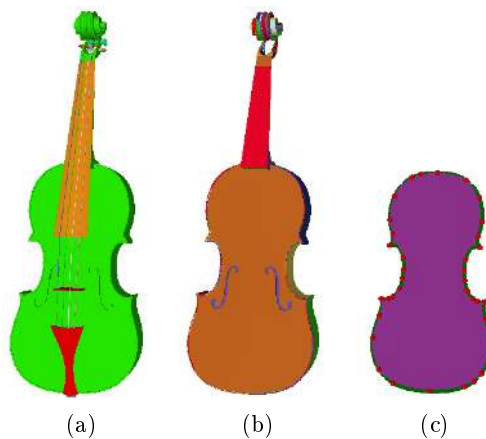


FIGURE 3.13 – Décomposition d'un modèle BRep-NURBS. (a) Les objets BRep dans un modèle représentant un violon. (b) L'objet BRep représentant la table d'harmonie et le manche du violon. (c) Les entités BRep d'une face.

le contour global de cette *face*. Nous utilisons alors les *loops* extérieures dans notre algorithme pour caractériser les faces.

Comme les topologies des modèles BRep-NURBS sont très variées, ces modèles peuvent avoir une ou plusieurs faces. Pour certains modèles qui ont plusieurs faces, les *loops* extérieures d'une *face* peuvent être utilisées comme une référence pour chercher les *faces* similaires et pour estimer les transformations. Mais elles ne sont plus applicables pour les autres modèles ayant une seule face. Pour ces derniers, l'approche de points discrets [Podolak 2006, Mitra 2006] est privilégiée à la détection. Alors, nous proposons deux méthodes dans l'étape d'extraction des isométries : l'une basée topologie et l'autre basée géométrie.

2.2 Algorithme proposé

Nous proposons un algorithme de détection des isométries partielles qui comprend plusieurs étapes. Pour généraliser cet algorithme, nous classifions ces étapes en quatre tâches principales : l'extraction des isométries, la classification des isométries, la validation et l'extension.

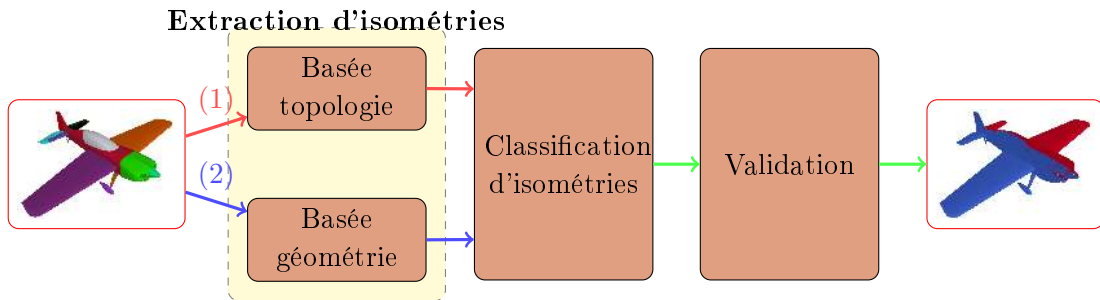


FIGURE 3.14 – Algorithme proposé.

Comme illustré dans la figure 3.14, notre algorithme propose deux processus avec traitements différents. Si le modèle BRep-NURBS en entrée possède plusieurs faces, il sera traité par le processus (1). Par contre, s'il possède seulement une face ou le processus (1) ne donne pas de résultat, il sera traité par le processus (2). Les deux processus ont les mêmes tâches principales. Les sections suivantes détaillent les tâches principales dans notre algorithme proposé.

3 Extraction des isométries

À l'entrée de l'algorithme, les *entités de référence* du modèle BRep-NURBS sont générées. Cette tâche comprend deux étapes principales : la mise en correspondance des entités de même signature et l'estimation de l'isométrie entre les paires d'entités. Comme nous considérons deux *entités de référence* différentes : les faces et les points, il s'agit de deux méthodes différentes pour la mise en correspondance et pour l'estimation des isométries. Ainsi, la signature d'une face est caractérisée par

sa boîte englobante et la topologie de sa *loop* extérieure. La signature d'un point de la surface NURBS est caractérisé par ses propriétés géométriques différentielles. Nous introduisons pour cette tâche deux méthodes différentes pour l'extraction des différentes isométries : l'extraction basée topologie et l'extraction basée géométrie.

3.1 Extraction basée topologie

Avant d'entrer dans les détails, nous introduisons quelques définitions nécessaires basées sur les entités des objets BRep.

3.1.1 Définitions

Étant données deux faces $F^i = \{S^i, V^i\}$ et $F^j = \{S^j, V^j\}$, supposons que les deux faces aient le même nombre de sommets.

Définition 15 F^i et F^j sont dites *topologiquement similaires* à une isométrie T^{ij} près ssi T^{ij} transforme V^i en V^j :

$$V^j = T^{ij}(V^i). \quad (3.5)$$

Définition 16 F^i et F^j sont dites *géométriquement similaires* à une isométrie T^{ij} près ssi T^{ij} transforme S^i en S^j :

$$S^j = T^{ij}(S^i). \quad (3.6)$$

Définition 17 F^i et F^j sont dites *similaires* à une isométrie T^{ij} près ssi elles sont similaires à la fois topologiquement et géométriquement par cette isométrie.

Remarque 2 Les définitions 15 et 16 sont indépendantes. Pendant que la définition 15 exprime la condition de la similarité de deux ensembles des sommets de deux faces, la définition 16 exprime celle de leurs surfaces NURBS sous-jacentes.

Ces définitions sont importantes pour la suite de notre algorithme car elles offrent des critères nécessaires et suffisants pour identifier les *faces* similaires. Nous constatons que les définitions 15 et 16 introduisent des conditions nécessaires mais pas suffisantes pour confirmer que deux faces sont similaires à une isométrie près. Pour la première définition, la condition où deux faces ont des contours similaires à une isométrie près n'assure pas la similarité entre elles, puisqu'il est possible que les surfaces sous-jacentes soient de formes différentes. De même pour la deuxième définition, la condition où les surfaces adjacentes sont similaires à une isométrie près n'assure pas la similarité entre les deux faces car les contours de ces faces sont probablement différents. Alors, la définition 17 introduit des conditions nécessaires et suffisantes pour que deux faces soient similaires à une isométrie près, c'est-à-dire,

deux faces doivent être définies par des surfaces adjacentes et des contours similaires à une isométrie près.

Pour vérifier si deux faces F^i et F^j sont similaires à une isométrie près, nous pouvons estimer tout d'abord l'isométrie entre les premières entités soit les surfaces adjacentes soit les contours correspondants, cette isométrie est ensuite validée sur les entités restantes. En général, l'estimation de l'isométrie ne peut se faire après que les premières entités aient validés l'équivalence. L'équivalence entre deux ensembles de sommets est plus accessible et plus efficace qu'entre deux surfaces paramétriques. En effet, l'équivalence entre deux ensembles de sommets peut être facilement validée par la comparaison des distances entre ces sommets. Par contre, la comparaison entre deux surfaces paramétriques exige plus de considérations. Par exemple, deux surfaces similaires peuvent être paramétrées différemment par l'insertion de nœuds ou par l'inversion des paramètres (u, v) . Alors, dans cet algorithme, nous prenons en compte les sommets de faces pour tester s'il existe une isométrie entre elles.

3.1.2 Méthode générale

Étant donné un modèle $M = \{F^i\}, i \in [0, n_F)$, l'algorithme essaie d'identifier au maximum les paires de faces similaires. Il se divise en deux étapes comme illustré dans la figure 3.15.

1. **Mise en correspondance des faces.** Les paires de faces sont mises en correspondance s'il existe une isométrie entre leurs sommets. Notons $\Lambda = \{P_k\}$ l'ensemble des paires de faces topologiquement similaires, avec $P_k = \{F_k^i, F_k^j\}$.
2. **Estimation des isométries.** Pour chaque paire P_k , la matrice R_k et le vecteur t_k sont estimés en optimisant un problème aux moindres carrés, tel que :

$$V_k^j = R_k * V_k^i + t_k.$$

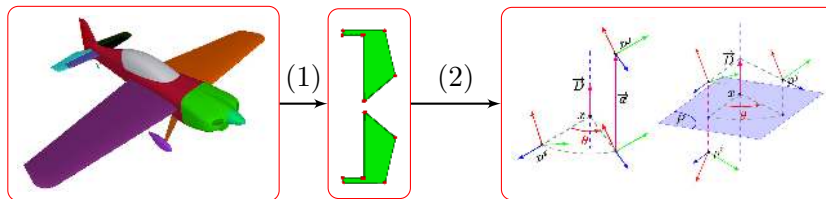


FIGURE 3.15 – Extraction d'isométries basée topologie.

Dans cette tâche d'extraction des isométries, l'objectif est d'identifier les paires de faces correspondances et d'estimer l'isométrie entre les faces de chaque paire. En fait, cette tâche estime l'isométrie entre deux ensembles de sommets de deux faces si elle existe. Dans la littérature, plusieurs recherches en 2D et 3D proposent des techniques pour résoudre ce problème.

Dans le domaine de l'imagerie, le *filtrage des points par motif* génère beaucoup de travaux avec plusieurs méthodes et techniques proposées. Ce problème est généralisé par l'exigence de mettre en correspondance deux images en 3D : l'une

appelée « image référencé » $P = \{(x_i, y_i, z_i), i = 1 \dots n_r\}$ et l'autre appelée « image perçu » $Q = \{(X_i, Y_i, Z_i), i = 1 \dots n_p\}$. La transformation affine estimée est de la forme :

$$\begin{aligned} X &= ax + by + cz + d \\ Y &= ex + fy + gz + h \\ Z &= ix + jy + kz + l \end{aligned} \quad (3.7)$$

Ici, les paramètres définis par les lettres alphabétiques de a à l sont à régler pour former la matrice de transformation :

$$T = \begin{bmatrix} a & b & c & d \\ e & f & g & h \\ i & j & k & l \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Dans un ouvrage, Goshtasby présente trois approches différentes pour mettre en correspondance des points et pour estimer la transformation affine entre ces points [Goshtasby 2005] :

- Cohérence de la scène : en supposant qu'il existe une correspondance entre quatre points non colinéaires dans les deux ensembles, un processus itératif vérifie la correspondance des points restants et rectifie la transformation affine correspondante. La distance de Hausdorff est utilisée pour mesurer la tolérance de la transformation estimée.
- Classification : pour chaque combinaison de quatre points, les paramètres de a à l sont estimés et définissent un vote dans l'espace des transformations. Le résultat final contient des paramètres dont les valeurs sont les plus récurrentes parmi les votes.
- Invariance : en considérant chaque point comme une pondération de quatre points non-colinéaires $p = p1 + \alpha_1 p2 + \alpha_2 p3 + \alpha_3 p4$, toutes les combinaisons de quatre points quelconques des deux images sont considérées afin d'identifier les deux combinaisons ayant les paramètres $(\alpha_1, \alpha_2, \alpha_3)$ les plus proches. Les paramètres de a à l sont estimés à partir de ces deux combinaisons de points.

Nous pouvons constater que ces approches sont des procédés itératifs pour définir les paramètres de la transformation affine, les résultats obtenus sont des solutions approchées. Pourtant, il n'y a aucune condition pour assurer la convergence de ces solutions. Ainsi, les transformations affines estimées ne sont pas tout à fait des isométries alors qu'il s'agit de l'objectif de nos premières étapes. De plus, toutes ces approches exigent que les ensembles aient au moins quatre points : ce n'est pas le cas pour les sommets des faces. Il faut donc procéder à une mise en correspondance des faces et à une estimation des isométries.

3.1.3 Mise en correspondance des faces

Étant données deux faces $F^i = \{S^i, V^i\}$ et $F^j = \{S^j, V^j\}$ où $V^i = \{v_0^i, \dots, v_{n-1}^i\}$ et $V^j = \{v_0^j, \dots, v_{m-1}^j\}$. L'étape de mise en correspondance commence par la mise en correspondance "un par un" des sommets de V^i et V^j où il existe une isométrie

préservant l'ordre des sommets dans les contours associés. Dans la plupart des cas, les faces similaires partagent le même nombre de sommets. Cependant, il est possible que des sommets supplémentaires aient été introduits pour raffiner les arêtes correspondantes. De ce fait, les faces similaires contiennent parfois un nombre différent de sommets bien que la topologie des contours soit la même. Pour limiter ce phénomène, nous ajoutons un filtrage des sommets de V^i et V^j , et considérons seulement les sommets dont les directions des tangentes à gauche et à droite sont assez différentes. Ce traitement nous permet de considérer seulement les "vrais" sommets. Nous avons besoin de considérer tous les sommets de départ possibles, par contre le sens de parcours est fixé. Dans le cas idéal, $n = m$, nous considérons donc toutes les correspondances entre les sommets de V^i et les sommets de V^j telles que :

$$\sum_{k=0}^{n-1} \left| \|v_k^i - v_{k+1}^i\| - \|v_{m \pm k[n]}^j - v_{m \pm (k+1)[n]}^j\| \right| \sim 0. \quad (3.8)$$

où $\|\bullet\|$ signifie la norme L_2 de la distance Euclidienne. Dans le cas où $m \neq n$, supposons que $n < m$, une étape supplémentaire détermine le sous-ensemble $V^o = \{v_0^o, \dots, v_{n-1}^o\} \subset V^j$ annulant l'expression suivante :

$$\sum_{k=0}^{n-1} \left| \|v_k^i - v_{k+1}^i\| - \|v_{n \pm k[n]}^o - v_{n \pm (k+1)[n]}^o\| \right|. \quad (3.9)$$

La correspondance entre ces sommets est juste une condition nécessaire, il faut valider la correspondance sur les surfaces sous-jacentes pour en déduire les faces similaires et l'isométrie associée.

3.1.4 Estimation des isométries

Étant donnés deux ensembles V^i et V^o dont les sommets sont mis en correspondance suivant l'équation (3.9), c'est-à-dire $v_0^i \mapsto v_0^o, \dots, v_{n-1}^i \mapsto v_{n-1}^o$. Cet étape vise à déterminer les paramètres de la transformation : la rotation R et la translation t , qui minimisent la quantité suivante :

$$e^2(R, t) = \frac{1}{n} \sum_{k=0}^{n-1} \|v_k^o - (R * v_k^i + t)\|^2. \quad (3.10)$$

L'équation 3.10 revient à résoudre un système linéaire du type $A \vec{x} = \vec{b}$, où A et \vec{b} sont respectivement la matrice et le vecteur composé par des coordonnées ordonnées de deux ensembles V^i et V^o , \vec{x} est la solution représentant les paramètres de la transformation. Nous résolvons donc ce système linéaire en appliquant la pseudo inverse, c'est-à-dire $\vec{x} = A^\dagger \vec{b}$ avec $A^\dagger = (A^T A)^{-1} A^T$.

Lorusso et al. [Lorusso 1997] comparent quatre algorithmes pour estimer les transformations. D'après leur comparaison, la méthode de *Décomposition en valeurs singulières* (SVD ou Singular Value Decomposition en anglais) d'une matrice est la plus efficace et la plus robuste. En effet, Arun et al. [Arun 1987] proposent

un algorithme basé sur la SVD d'une matrice 3×3 . Umeyama [Umeyama 1991] a proposé une amélioration de cet algorithme qui résout aussi les cas spéciaux (points coplanaires, transformation directe et indirecte).

Méthode de Décomposition en Valeurs Singulières

Dans cette étape, nous appliquons l'algorithme d'Umeyama [Umeyama 1991] pour estimer les transformations. Notons que cet algorithme estime uniquement une transformation directe et/ou une transformation indirecte si le nombre des sommets de chaque ensemble est supérieur à 2 ($n > 2$), et si les sommets de V^i (et V^o) ne sont pas colinéaires. Si V^i et V^o ne sont pas coplanaires, il existe une seule transformation. Par contre, s'ils sont coplanaires, il existe deux transformations différentes : l'une directe et l'autre indirecte. D'après l'équation (3.9), les transformations estimées préservent la longueur, et d'après les propriétés de la SVD, les parties linéaires associées sont orthogonales. Ces transformations sont donc des *isométries*. Cette étape donne au maximum deux isométries qui transforment V^i en V^o ; chaque isométrie est de la forme :

$$T^{ij} = \begin{bmatrix} R^{ij} & t \\ 0 & 1 \end{bmatrix}. \quad (3.11)$$

L'algorithme d'estimation de la transformation peut être résumé comme suit :

1. Formuler la matrice de covariance :

$$H = \sum_{k=0}^{n-1} (v_k^i - \mu_i)(v_k^o - \mu_o)^T. \quad (3.12)$$

où l'exposant T signifie la transposée de la matrice, μ_i et μ_o sont les deux barycentres de V^i et V^o .

2. Décomposer H en valeurs principales par SVD

$$H = USV^T \quad (3.13)$$

3. Calculer la matrice de rotation et le vecteur de translation correspondants :

$$R = VU^T \quad t = \mu_o - R * \mu_i. \quad (3.14)$$

D'après [Arun 1987], si l'une des valeurs singulières de S s'annule (équation (3.13)), c'est-à-dire $\lambda_1 > \lambda_2 > \lambda_3 = 0$, nous pouvons formuler une autre solution R' et t' en introduisant la matrice $V' = [v_1, v_2, -v_3]$, où v_1, v_2, v_3 sont les trois vecteurs colonnes de V correspondant aux trois valeurs singulières. Nous avons :

$$R' = V'U^T, \quad t' = \mu_o - R' * \mu_i. \quad (3.15)$$

Notons que l'isométrie estimée est directe si $\det(R) = 1$ et indirecte si $\det(R) = -1$. Il est possible que la transformation trouvée ne soit pas une isométrie puisque l'équation 3.9 est seulement une condition nécessaire (par exemple, non conservation des angles). Dans ce cas, $|\det(R)| \neq 1$ et on ne considère pas cette transformation.

3.2 Extraction basée géométrie

Cette méthode est adaptée aux modèles BRep-NURBS ayant une seule face où les isométries ne peuvent pas être identifiées par la méthode précédente. Considérons S , une surface NURBS en produit tensoriel de bidegré (p, q) associée aux vecteurs de noeuds $u = \{u_0, \dots, u_n\}$ et $\{v_0, \dots, v_m\}$, au réseau de points de contrôle $C = \{P_{ij} \mid i \in [0, n-p], j \in [0, m-q]\}$ pondérés par des poids $w_{ij} \in \mathbb{R}$ et aux fonctions de base N_i^d de degré $d \geq 0$, définie par l'équation suivante :

$$S(u, v) = \frac{\sum_{i=0}^{n-p} \sum_{j=0}^{m-q} N_i^p(u) N_j^q(v) w_{ij} P_{ij}}{\sum_{i=0}^{n-p} \sum_{j=0}^{m-q} N_i^p(u) N_j^q(v) w_{ij}}. \quad (3.16)$$

Dans cette méthode d'extraction, une grille de points de la surface est obtenue en échantillonnant les paramètres du modèle en u et v . Puisque les positions des points ne sont pas suffisantes pour estimer la transformation euclidienne, leurs propriétés géométriques sont aussi considérées. A chaque point de la surface, il peut être associé des propriétés différentielles locales de la surface parmi lesquelles des courbures et des directions (tangentes) principales de courbure.

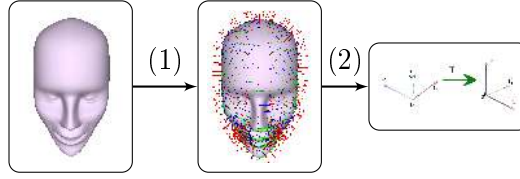


FIGURE 3.16 – Extraction des isométries à partir d'une grille de points de la surface.

3.2.1 Signatures des points de la surface paramétrique

Chaque point de la surface NURBS a une *signature* composée des courbures principales (κ_1, κ_2) et du *repère local* des directions principales \vec{t}_1, \vec{t}_2 et de la normale \vec{n} . Dans les surfaces paramétriques, les directions des normales et des tangentes sont orientées par la paramétrisation, nous pouvons donc définir des repères locaux orthonormés tels que $\vec{n} = \vec{t}_1 \otimes \vec{t}_2$.

Le calcul des signatures d'un point spécifique sur une surface B-Spline ou NURBS est basé sur les propriétés locales différentielles ; leur calcul pour les surfaces NURBS est décrit par Farin [Farin 1992]. La *première forme fondamentale* est définie par :

$$ds^2 = E du^2 + 2F du dv + G dv^2 \quad (3.17)$$

$$\text{où } \begin{cases} E = E(u, v) = \chi_u \chi_u \\ F = F(u, v) = \chi_u \chi_v \\ G = G(u, v) = \chi_v \chi_v \end{cases}$$

Le discriminant est défini par :

$$D = \|\chi_u \wedge \chi_v\| = \sqrt{EG - F^2} \quad (3.18)$$

Dans l'équation (3.17), les dérivées partielles χ_u et χ_v en un point \mathbf{x} engendrent le plan tangent à la surface en \mathbf{x} . Alors, le vecteur unitaire normal

$$n = \frac{\chi_u \wedge \chi_v}{\|\chi_u \wedge \chi_v\|} = \frac{1}{D} (\chi_u \wedge \chi_v) \quad (3.19)$$

avec les vecteurs non normalisés χ_u, χ_v , former un repère affine d'origine \mathbf{x} .

La courbure d'une région de la surface est basée sur la *deuxième forme fondamentale* :

$$\kappa \cos \phi ds^2 = L du^2 + 2M dudv + N dv^2 \quad (3.20)$$

$$\text{avec } \begin{cases} L = \chi_{uu} \cdot n \\ M = \chi_{uv} \cdot n \\ N = \chi_{vv} \cdot n \end{cases}$$

où χ_{uu}, χ_{uv} et χ_{vv} sont les dérivées partielles secondes en un point \mathbf{x} .

L'équation (3.20) signifie que, pour une direction donnée du/dv dans le plan u, v et pour un angle donné ϕ , la *deuxième forme fondamentale*, avec la *première forme fondamentale*, nous permettent de calculer la courbure κ d'une courbe tracée sur une surface dont la tangente a cette direction. Nous introduisons deux matrices symétriques :

$$\mathcal{F}_1 = \begin{pmatrix} E & F \\ F & G \end{pmatrix} \text{ et } \mathcal{F}_2 = \begin{pmatrix} L & M \\ M & N \end{pmatrix} \quad (3.21)$$

Notation 2 Puisque χ_u et χ_v sont linéairement indépendantes, \mathcal{F}_1 est toujours inversible. De plus, la matrice $\mathcal{F}_1^{-1} \mathcal{F}_2$ possède toujours des valeurs propres réelles. Les deux valeurs propres κ_1, κ_2 sont les deux courbures principales et les deux vecteurs propres $V_1 = (\xi_1, \eta_1)^T, V_2 = (\xi_2, \eta_2)^T$ définissent les deux directions principales $\vec{t}_1 = \xi_1 \vec{\chi}_u + \eta_1 \vec{\chi}_v, \vec{t}_2 = \xi_2 \vec{\chi}_u + \eta_2 \vec{\chi}_v$.

Théorème 1 Etant données κ_1 et κ_2 les courbures principales en un point \mathbf{p} d'une région de la surface χ . Alors :

- $\kappa_1, \kappa_2 \in \mathbb{R}$
- si $\kappa_1 = \kappa_2 = \kappa$, toutes les tangentes à p sont des vecteurs principaux, et le point p est appelé point ombilic.
- si $\kappa_1 \neq \kappa_2$, les deux vecteurs principaux correspondants sont perpendiculaires l'un à l'autre.

La signature en chaque point correspond à un repère orthonormé affine dont l'origine est ce point et les vecteurs sont le vecteur normal et les deux vecteurs principaux. Deux de ces repères orthonormés permettent de caractériser l'isométrie faisant passer de l'un à l'autre. Dans la suite, nous appellerons surfaces similaires deux régions de la surface similaires à une isométrie près. Dans le cas des points ombilics, les directions des axes sont mal définies, et donc nous ne considérerons

pas ces points particuliers. Pour les autres points (non ombilics), les directions du repère sont bien définies par la paramétrisation. Dans le paragraphe suivant, nous expliquons comment nous déterminons le sens des vecteurs et le calcul de la transformation quelque peu différents de celui proposé par Mitra et al. [Mitra 2006].

3.2.2 Orientation dans la quadrille paramétrique

Parmi les points échantillonnés, les couples de points ayant la même signature sont mis en correspondance pour évaluer les transformations entre eux. Étant donnée une paire de points (p_i, p_j) de même signature, la transformation de p_i à p_j est définie par $T_{ij} \in \mathbb{R}^6$, dont $T_{ij} = (R_{ij}^x, R_{ij}^y, R_{ij}^z, t_{ij}^x, t_{ij}^y, t_{ij}^z)$, où $[R_{ij}^x, R_{ij}^y, R_{ij}^z]$ sont les trois angles d'Euler dérivés de la matrice de rotation R_{ij} et $[t_{ij}^x, t_{ij}^y, t_{ij}^z]$ sont les trois composantes de la translation définie par $t_{ij} = p_j - R_{ij} * p_i$. La matrice de rotation R_{ij} est calculée à partir des deux repères orthonormaux conformément à la règle de la main droite (voir figure 3.17). Cette matrice est évaluée afin que le repère local de p^i soit confondu avec celui de p^j :

$$R_{ij} = \begin{bmatrix} \vec{n}^i \\ \vec{t}_1^i \\ \vec{t}_2^i \end{bmatrix}^T * \begin{bmatrix} \vec{n}^i \cdot \vec{n}^j & \vec{n}^i \cdot \vec{t}_1^j & \vec{n}^i \cdot \vec{t}_2^j \\ \vec{t}_1^i \cdot \vec{n}^j & \vec{t}_1^i \cdot \vec{t}_1^j & \vec{t}_1^i \cdot \vec{t}_2^j \\ \vec{t}_2^i \cdot \vec{n}^j & \vec{t}_2^i \cdot \vec{t}_1^j & \vec{t}_2^i \cdot \vec{t}_2^j \end{bmatrix} * \begin{bmatrix} \vec{n}^j \\ \vec{t}_1^j \\ \vec{t}_2^j \end{bmatrix} \quad (3.22)$$

En effet, l'équation (3.22) exprime R_{ij} comme un changement de base du repère local de p_i à celui de p_j .

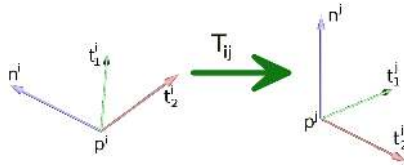


FIGURE 3.17 – Transformation entre deux points p^i et p^j .

Chacune des paires de points (p^i, p^j) correspond à une transformation Euclidienne T_{ij} ; on considère ensuite l'espace Γ contenant ces transformations.

3.2.3 Analyse du calcul direct des transformations

On constate que les caractéristiques choisies pour les transformations jouent un rôle primordial sur tout l'algorithme car elles sont leurs coordonnées dans l'espace des transformations. Donc, il est important que le calcul de ces transformations soit *fiable et efficace*.

En effet, à chaque point de la surface paramétrique, l'orientation des vecteurs du repère local est bien définie et déterminée par défaut par la paramétrisation. Comme le montre la figure 3.18, des tangentes maximales (en bleu) et des tangentes minimales (en violet) sont orientées naturellement vers les mêmes directions. Ainsi,

les normales sont bien définies grâce à la paramétrisation. En fait, d'après l'équation (3.22), les transformations évaluées entre les deux repères locaux quelconques correspondent à des transformations directes.

En pratique, considérer seulement les transformations directes n'est pas suffisant ; par exemple les symétries planaires ne peuvent pas être identifiées. Pour mieux comprendre le problème, considérons maintenant la figure 3.19 qui illustre un exemple de l'évaluation des transformations d'une surface NURBS ayant deux régions symétriques. La figure 3.19b présente une vue simplifiée des neuf paires de points de mêmes signatures qui sont connectées par les lignes de différentes couleurs. Suivant le *procédé de transformée de Hough* [Mitra 2006], ces paires doivent partager une même transformation. La figure 3.19c montre des résultats contredisant cette idée. La transformation d'une paire aléatoire parmi elles est estimée et appliquée aux points p_i (à droite) des autres paires. On trouve que les points calculés ne sont pas confondus avec les points p_j des autres paires. En effet, il n'est pas convenable qu'une transformation directe soit identifiée sur des points symétriques.

L'évaluation des transformations entre les repères (p^i, n^i, t_1^i, t_2^i) et (p^j, n^j, t_1^j, t_2^j) n'est pas cohérente, en particulier lorsqu'on considère les transformations indirectes. La figure 3.20 présente deux repères locaux qui appartiennent à deux parties symétriques mais qui ne sont pas cohérents (la tangente minimale du repère droit doit être orientée dans l'autre sens).

Une modification de l'orientation des repères locaux est nécessaire. Considérons maintenant deux points de la surface avec leurs repères locaux illustrés dans la figure 3.21a. Les vecteurs normaux sont bien orientés par la paramétrisation, nous allons donc modifier la direction des vecteurs tangents pour obtenir une représentation cohérente de ces repères.

Supposons que le repère de référence à gauche de la figure 3.21a soit fixé (les directions principales sont fixées arbitrairement), nous pouvons observer qu'il y a quatre possibilités d'orientation des directions principales du repère de droite, c'est-à-dire, il faut choisir le meilleur parmi ces quatre repères.

Pour choisir le meilleur repère, nous projetons les points voisins sur le plan de ses

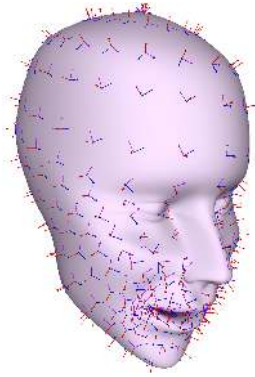


FIGURE 3.18 – Orientation des tangentes principales.

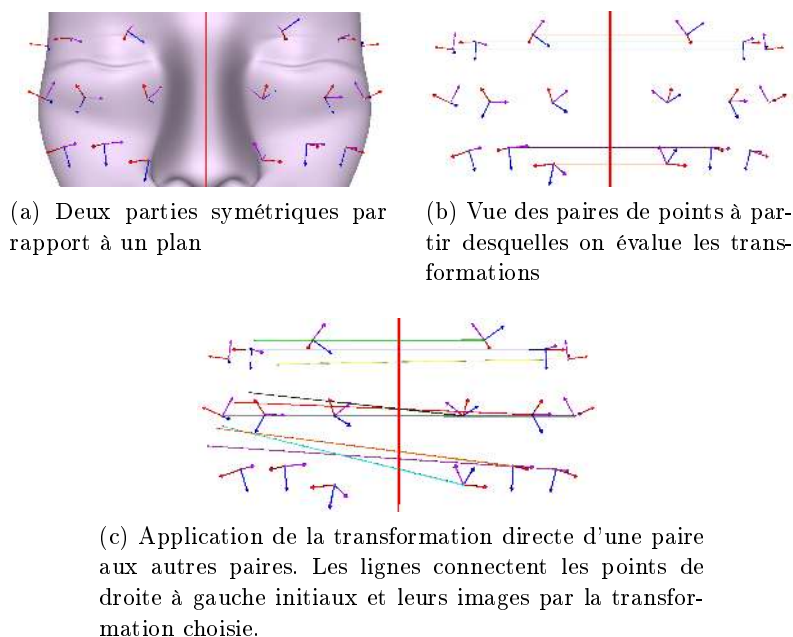


FIGURE 3.19 – Exemple sur l'évaluation des caractéristiques de transformations.

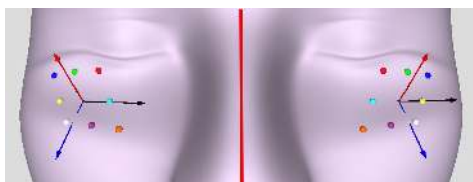


FIGURE 3.20 – Incohérence entre deux repères locaux symétriques.

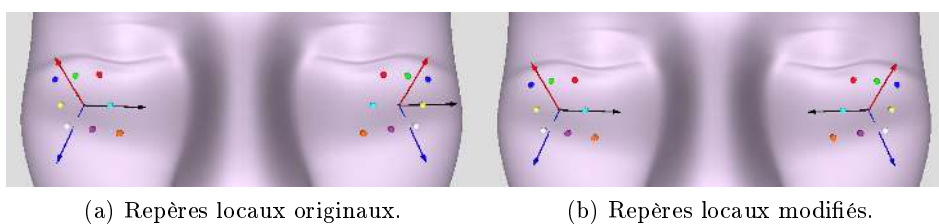


FIGURE 3.21 – Réorientation par le voisinage.

vecteurs tangents. Pour chaque repère, ses voisins sont ordonnés en tournant le repère tangentiel autour de son origine. Nous obtenons alors quatre suites de courbures principales de ses voisins. Ces suites sont alors comparées avec celles du repère de référence pour choisir la suite la plus convenable. Il exige donc d'effectuer plusieurs tests pour trouver une bonne correspondance entre les deux suites. Pour améliorer la performance du traitement, nous adaptons l'algorithme Goodman-Kruskal [Campello 2009] qui est très efficace dans la mise en correspondance de deux suites

de valeurs réelles.

La figure 3.21b représente le résultat de la modification de l'orientation par le voisinage. Cette modification nous permet d'évaluer de manière cohérente la transformation entre les repères locaux.

Cette modification nous fournit ainsi un moyen efficace pour différencier les transformations directes et indirectes. De plus, chacune de ces transformations affines est associée à une application vectorielle linéaire qui appartient au groupe orthogonal dans l'espace à trois dimensions. Ces transformations sont donc des isométries.

3.3 Caractérisation des isométries

Comme nous l'avons décrit dans la section 3.1.2, les isométries sont caractérisées en fonction de la nature de leurs points fixes. Dans ce paragraphe, nous résumons la caractérisation d'une isométrie à travers l'algorithme suivant :

Algorithme 5 Caractérisation d'une isométrie.

Précondition : Une isométrie f .

```

Décomposer  $f = g \circ t_{\vec{a}}$ .
Évaluer la matrice de rotation  $R$  de  $g$ .
si  $\det(R) == 1$  alors
    si  $g$  a une ligne de points fixes alors
        si  $t_{\vec{a}} == \emptyset$  alors
             $f = r(D, \theta)$ .
        sinon
             $f = r(D, \theta) \circ t_{\vec{a}}$ .
    si  $g$  a un espace de points fixes alors
        si  $t_{\vec{a}} == \emptyset$  alors
             $f$  est Identité.
        sinon
             $f = t_{\vec{a}}$ .
    sinon
        si  $g$  a un seul point fixe alors
             $f = s(P) \circ r(D, \theta)$ .
        si  $g$  a un plan de points fixes alors
            si  $t_{\vec{a}} == \emptyset$  alors
                 $f = s(P)$ .
            sinon
                 $f = s(P) \circ t_{\vec{a}}$ .
retour  $f$ 

```

3.4 Conclusion

À la fin de cette tâche, nous avons un ensemble des paires d'*entités de référence*, soit des faces soit des points, et les isométries correspondantes qui lient les deux entités dans chaque paire. Cependant, les isométries sont différentes de l'une à l'autre et la cohérence géométrique entre les entités est perdue car elles sont positionnées partout dans le modèle. Alors, il faut classer les paires des entités dont les isométries ont les mêmes caractéristiques. Ainsi, pour obtenir les régions cohérentes d'un groupe d'isométries, il faut regrouper les entités proches et étendre vers les entités qui ne sont pas identifiées par cette tâche mais qui peuvent contribuer au résultat final. Nous allons introduire ces tâches dans les sections suivantes.

Pour la classification des isométries, la caractérisation des isométries (présentée précédemment) est très avantageuse. En effet, les isométries sont identifiées par les classes d'isométries (section 3.1.2) : la translation, la rotation, la symétrie ou une composition de ces trois dernières. La complexité de l'espace des isométries est réduite en introduisant six sous-espaces différents. Chaque sous-espace comprend des isométries de même classe, la classification de ces isométries devient plus facile et efficace.

4 Classification des isométries

Dans un espace des isométries Γ de dimension d^Γ , les isométries qui partagent les mêmes caractéristiques sont regroupées. Suivant la complexité des modèles, la classification des isométries peut radicalement différer d'un modèle à un autre. Pour cela, des méthodes de classification non supervisées sont privilégiées. Elles consistent à partitionner un ensemble de données de Γ en \hat{k} classes de manière à ce que les classes soient les plus compactes possibles avec un minimum de connaissances a priori quant au nombre et à la forme des classes. Mitra et al. [Mitra 2012] proposent l'utilisation de la méthode Mean Shift [Comaniciu 1999, Comaniciu 2002] pour partitionner cet ensemble de données. Dans la suite, nous présentons cette méthode et proposerons une autre méthode de classification appelée classification spectrale.

4.1 Méthode Mean Shift

Cette méthode est basée sur l'estimation de densité [Parzen 1962, Rosenblatt 1956]. Le but de cette dernière consiste à estimer les paramètres de la loi de probabilité P qui est décrite par un vecteur des isométries $\{x_i\}_{i=1..n}$ dans Γ . Cette approche est dite non-paramétrique car aucune connaissance a priori est requise quant à la loi de probabilité P . On suppose donc que le nombre de paramètres qui décrivent la loi P est une fonction croissante du nombre de données, ou encore que le nombre de paramètres est infini. L'estimation de densité est réalisée à l'aide d'une fonction noyau définie comme suit :

Définition 18 Une fonction k de deux variables est dite fonction noyau si elle est symétrique positive :

- $\forall x, y, k(x, y) = k(y, x)$
- $\forall a_1 \dots a_n \in \mathbb{R}$ et les vecteurs $x_1 \dots x_n \in \mathbb{R}^p$, $\sum a_i a_j k(x_i, x_j) \geq 0$

On dira que la matrice noyau de n vecteurs est la matrice symétrique positive associée au noyau :

$$k = [k(x_i, x_j)]_{i,j} \in \mathcal{S}_+^n$$

où $\mathcal{S}_+^n = \{M | \forall a \in \mathbb{R}^n, a^T M a \geq 0\}$ est l'ensemble des matrices symétriques positives.

L'estimation de la densité P revient à lisser, par une fonction noyau k , l'histogramme des données $\{x_i\}_{i=1..n}$ via une fenêtre glissante h :

$$P(x) \equiv \frac{1}{nh^p} \sum_{i=1}^n K\left(\frac{x - x_i}{h}\right). \quad (3.23)$$

où $K(x) = c_{k,p} k(x, x)$ avec $c_{k,p}$ est une constante de normalisation de la fonction noyau k telle que $0 \leq K(x) \leq 1$. Le membre de droite est appelé estimateur à noyau.

La méthode *Mean Shift* consiste alors à chercher la position qui, localement, possède la densité la plus élevée [Comaniciu 1999, Comaniciu 2002]. Elle revient donc à modéliser le gradient de densité :

$$\nabla P(x) = \frac{d}{dx} \left(\frac{1}{nh^p} \sum_{i=1}^n K\left(\frac{x - x_i}{h}\right) \right). \quad (3.24)$$

Pour calculer ∇P , il faut introduire la fonction g , la dérivée de la fonction noyau k , définie par :

$$g(x, x) = -k'(x, x), \forall x \in [0, \infty[$$

et le noyau

$$G(x) = c_{g,d} g(x, x)$$

avec $c_{g,d}$ une constante de normalisation telle que $0 \leq G(x) \leq 1$.

$$\begin{aligned} \nabla P(x) &= \frac{2c_{k,d}}{nh^{d+2}c_{g,d}} \sum_{i=1}^n (x_i - x) G\left(\frac{x - x_i}{h}\right) \\ &= \frac{2c_{k,d}}{nh^{d+2}c_{g,d}} \left[\sum_{i=1}^n G\left(\frac{x - x_i}{h}\right) \right] \left[\frac{\sum_{i=1}^n x_i G\left(\frac{x - x_i}{h}\right)}{\sum_{i=1}^n G\left(\frac{x - x_i}{h}\right)} - x \right] \end{aligned} \quad (3.25)$$

où $\sum_{i=1}^n G\left(\frac{x - x_i}{h}\right)$ est supposé strictement positif.

Le premier terme est proportionnel à la densité estimée au point x calculée avec le noyau $G(x)$. Le second terme, aussi appelé vecteur *mean shift* et noté m_h , est la différence entre la moyenne pondérée par le noyau G et x , le centre du noyau (ou de la fenêtre) :

$$m_h(x) = \frac{\sum_{i=1}^n x_i G\left(\frac{x - x_i}{h}\right)}{\sum_{i=1}^n G\left(\frac{x - x_i}{h}\right)} - x$$

Ce vecteur est toujours dans la direction de la densité maximale [Comaniciu 2002]. L'algorithme Mean Shift, à l'itération t , est composée de deux étapes :

- le calcul de la densité locale en un point m_h ;
- le déplacement itératif de la fenêtre en direction du gradient de densité maximum : $x^{t+1} = x^t + m_h$.

Ces étapes sont itérées jusqu'à converger vers un point où le gradient de la fonction densité est nulle c'est-à-dire vers le mode local de la densité. La classification finale des données regroupe les points ayant convergé vers le même mode. La figure 3.22 illustre l'algorithme de la méthode Mean Shift sur un exemple de deux distributions de données [Comaniciu 1999] : les calculs successifs du vecteur *mean shift* définissent deux chemins convergeant vers les deux densités locales maximales.

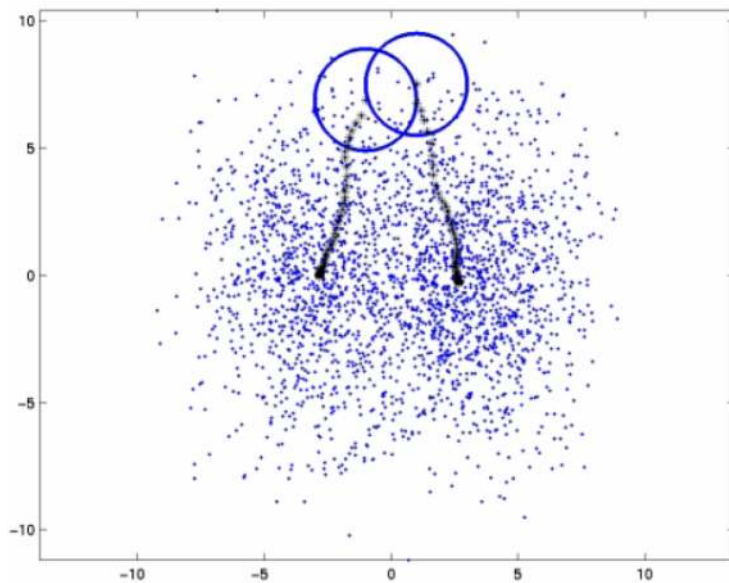


FIGURE 3.22 – Principe de la méthode Mean Shift [Comaniciu 1999].

Cette méthode présente l'avantage de fournir automatiquement le nombre de classes. Mais elles sont basées sur l'estimation d'une densité à noyau dont la taille h doit être définie au préalable. La principale difficulté de cette méthode est donc la forte sensibilité de la largeur de la fenêtre h , c'est-à-dire le choix de la taille du noyau. Les méthodes *mean shift* usuelles utilisent un noyau de taille fixe pour tout l'espace des données. Mitra et al [Mitra 2006] utilise le noyau Epanechnikov symétrique, $K(u) = \frac{3}{4}(1 - u^2)\mathbb{I}_{|u| \leq 1}$ et définit le paramètre h en équilibrant le biais et la variance de l'estimateur [Comaniciu 2003].

Cependant, lorsque les données varient de façon significative dans l'espace des transformations Γ , les méthodes *mean shift* deviennent très sensibles et peuvent donner de mauvais résultats. Nous proposons dans la suite une autre méthode de classification non supervisée.

Remarque 3 Des noyaux à tailles variables ont été proposés en définissant une sélection de tailles prédéfinies et en choisissant celles qui fournissent la meilleure approximation locale des données multidimensionnelles par une loi normale [Comaniciu 2003].

Mais lorsque les modèles sont complexes, l'ensemble des tailles prédéfinies devient grand et les algorithmes deviennent coûteux.

4.2 Méthode Spectrale

La classification spectrale n'utilise pas la fonction noyau pour estimer une densité de probabilité mais pour définir une mesure de similarité (ou affinité) entre les données. Il s'agit d'une méthode à noyau c'est-à-dire un algorithme prenant en entrée une matrice noyau. Le but revient donc à partitionner un ensemble de données en \hat{k} classes de manière à ce que des points qui appartiennent à la même classe soient le plus semblable possible et des points appartenant à des classes différentes soient le moins semblable possible.

Introduite dans le domaine de l'apprentissage automatique par *Shi et al.* [Shi 2000], *Ng et al.* [Ng 2002] et *Meila et Shi* [Meila 2001], la méthode de classification spectrale consiste à extraire les vecteurs propres associés aux plus grandes valeurs propres d'une matrice noyau gaussien normalisé [Von Luxburg 2007]. Ces vecteurs propres constituent un espace propre de dimension réduite dans lequel les données projetées sont regroupées en classes. L'algorithme associé à cette méthode est décrit ci-dessous (Algorithme 6).

Algorithme 6 Classification spectrale

Input : Ensemble $S = \{x_i\}_{i=1..n} \in \Gamma$, nombre de classes \hat{k} .

1. Construction de la matrice affinité $A \in \mathbb{R}^{n \times n}$ définie par :

$$A_{ij} = \begin{cases} \exp\left(-\frac{\|x_i - x_j\|^2}{(\sigma/2)^2}\right) & \text{si } i \neq j, \\ 0 & \text{sinon.} \end{cases} \quad (3.26)$$

où $\|\mathbf{x}\|$ est une métrique prédéfinie.

2. Construction de la matrice normalisée :
 $L = D^{-\frac{1}{2}} A D^{-\frac{1}{2}}$ avec D une matrice diagonale d'éléments $D_{i,i} = \sum_{r=1}^n A_{ir}, \forall i \in \{1, ..n\}$.
 3. Assembler la matrice $X = [X_1 X_2 .. X_{\hat{k}}] \in \mathbb{R}^{n \times \hat{k}}$ formée à partir des \hat{k} plus grands vecteurs propres de L .
 4. Construction de la matrice Y formée des lignes de X normalisées L^2 .
 5. Traiter chaque ligne de Y comme un point de $\mathbb{R}^{\hat{k}}$ et les classer en \hat{k} classes via la méthode *K-means*.
 6. Assigner le point original x_i à la classe θ si et seulement si la ligne i de la matrice Y est assignée à la classe θ .
-

Ng et al. [Ng 2002] justifie cet algorithme en considérant un cas géométrique idéal avec trois classes bien séparées (Figure 3.23 (a)). En partant de l'hypothèse que les points sont déjà ordonnés consécutivement par classe, la matrice affinité bien que

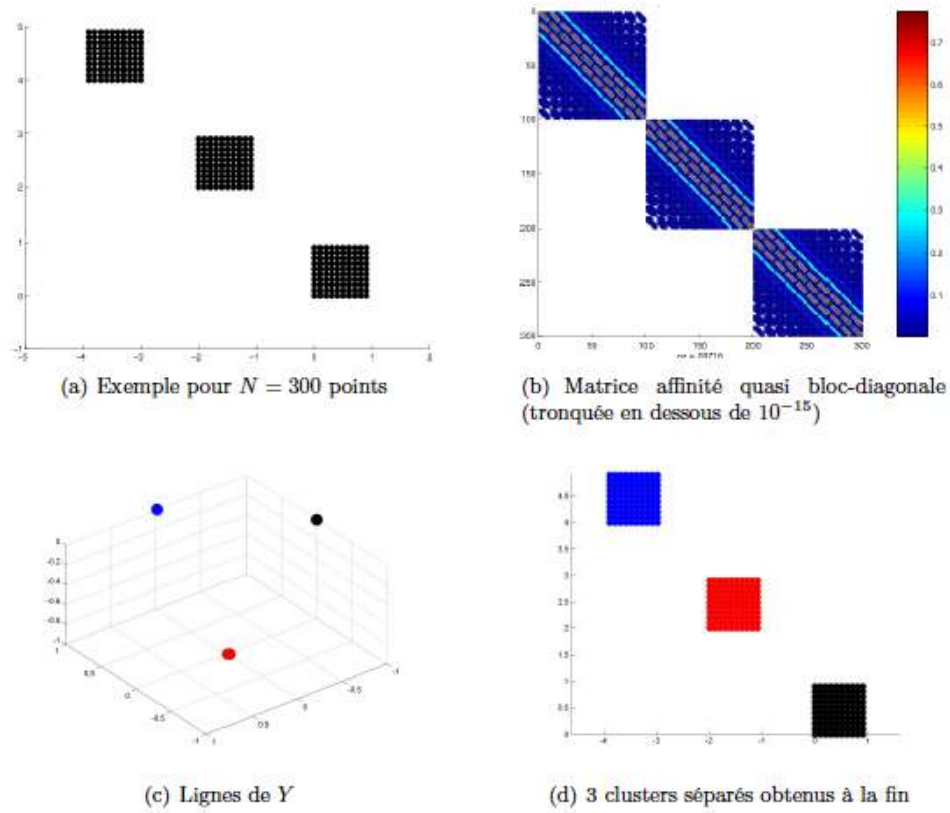


FIGURE 3.23 – Illustration des étapes de la classification spectrale.

dense, a alors une structure numérique proche de celle d'une matrice bloc-diagonale (Figure 3.23 (b)). Ainsi, la plus grande valeur propre de la matrice gaussienne normalisée est 1 avec un ordre de multiplicité égal à 3. Les lignes normalisées de la matrice constituée des vecteurs propres associés aux plus grandes valeurs propres sont donc constantes par morceaux. Dans l'espace défini par les $\hat{k} = 3$ plus grands vecteurs propres, il est facile alors d'identifier les 3 points compacts séparés qui correspondent aux 3 constantes par morceaux des vecteurs propres (Figure 3.23 (c)). Ainsi en appliquant la méthode *K-means* pour $\hat{k} = 3$, la classification est réalisée dans l'espace de projection spectrale : chaque point correspond à une classe. Enfin, par équivalence, la classification dans l'espace original est directement obtenue : chaque point correspond à un bloc (Figure 3.23 (d)).

Cette méthode est principalement basée sur la mesure d'affinité gaussienne, son paramètre et ses éléments spectraux. Elle présente l'intérêt d'utiliser les propriétés inhérentes aux noyaux de Mercer (ici, le noyau gaussien) permettant de projeter implicitement les données dans un espace de grande dimension dans lequel les données seront linéairement séparables. Ainsi, des classes de formes arbitraires (notamment des domaines non convexes) peuvent être définies [Von Luxburg 2007].

L'algorithme associé comprend peu d'étapes et peut être facilement codé sur

Matlab. De plus, l'algorithme dépend uniquement de deux paramètres à savoir le paramètre d'affinité gaussienne et le nombre de classes \hat{k} . Pour rendre cette méthode totalement non supervisée, nous utilisons une heuristique pour définir chaque paramètre [Mouysset 2010].

Paramètre d'affinité L'expression de l'affinité gaussienne, définie par l'équation (3.26), dépend donc d'un paramètre σ . Or, σ influe sur la séparabilité des données dans l'espace de projection spectrale. En effet, d'après Ng et al. [Ng 2002], σ contrôle la similarité entre les données et conditionne la qualité des résultats. Pour garder l'efficacité de la méthode, une approche globale est privilégiée.

Comme le paramètre σ sert de seuil sur les distances entre isométries, ce dernier doit être alors fonction des distances entre les points et de la dimension du problème. On se base pour le définir sur le cas d'une distribution uniforme où tous les points sont équidistants les uns des autres.

Soit $S = \{x_i\}_{i=1..n}$ un ensemble de n points dans l'espace des isométries Γ dont la distribution est isotropique. Les éléments de S sont inclus dans une boîte carrée englobante de dimension d^Γ et de longueur de côté $D_{max} = \max_{i \neq j} \|x_i - x_j\|$. En divisant cette boîte en n volumes identiques, une distance uniforme, notée D_{unif} , est définie :

$$D_{unif} = \frac{\max_{i \neq j} \|x_i - x_j\|}{n^{1/d^\Gamma}}. \quad (3.27)$$

Donc on peut considérer que s'il existe des classes alors il existe des points qui sont séparés d'une distance inférieure à D_{unif} . Ainsi, le paramètre gaussien σ est pris égal à une fraction de la distance D_{unif} :

$$\sigma = \frac{D_{unif}}{2}. \quad (3.28)$$

Cette heuristique intègre donc la notion de densité de points dans l'ensemble des données de dimension d^Γ et l'équation (3.28) donne un seuil à partir duquel des points sont considérés comme proches. La figure 3.24 représente le principe de l'heuristique sur un exemple 2D.

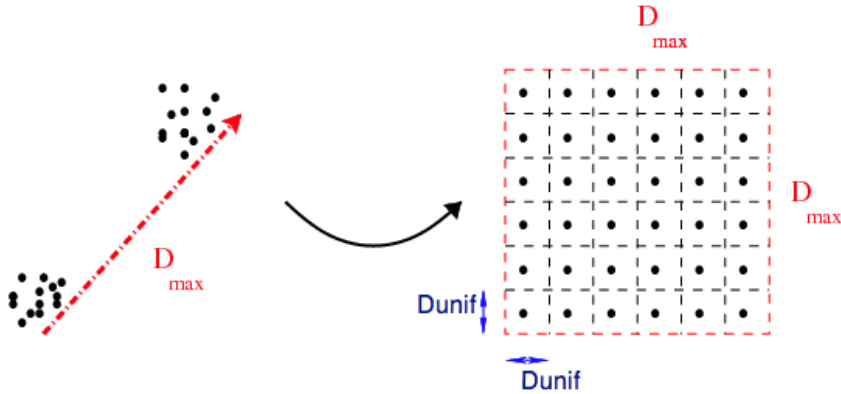


FIGURE 3.24 – Principe de l'heuristique (3.27).

Nombre de classes Le problème du choix du nombre de classes est un problème général pour les algorithmes de classification non supervisée. Pour déterminer ce nombre de classes \hat{k} , la matrice affinité gaussienne A est à nouveau exploitée et une mesure de qualité basée sur des ratios de normes de Frobenius est définie.

Soit α_k un nombre limite de classes à chercher. Pour une valeur $k' \in [2, \alpha_k]$, la matrice affinité est réordonnancée suivant les k' classes. Une matrice par bloc est ainsi définie : les blocs hors diagonaux représentent les affinités entre les classes et les blocs diagonaux les affinités intra-classes. A partir de cette structure bloc, nous pouvons évaluer un ratio moyen, noté $r_{k'}$, entre tous les blocs hors diagonaux et les blocs diagonaux en norme de Frobenius. Ainsi, parmi les valeurs de $k' \in [2, \alpha_k]$, le minimum de ce ratio correspond au cas où l'affinité entre les classes est la plus faible et l'affinité au sein des classes est la plus forte et définit le nombre optimal de classes \hat{k} :

$$\hat{k} = \arg \min_{k' \in [2, \alpha_k]} r_{k'}. \quad (3.29)$$

En effet, plus ce ratio est proche de 0, plus la matrice réordonnancée aura une structure bloc diagonale, comme représentée sur la figure 3.23 (b) avec l'exemple des trois blocs (figure (a)). Ainsi, on obtient une partition où les classes sont séparées par une distance suffisante permettant d'avoir des vecteurs propres constants par morceaux. Les données projetées dans l'espace spectral sont concentrées par classes et séparées comme le montre la figure 3.23 (c). La figure (b) représente les valeurs du ratio $r_{k'}$ en fonction des valeurs de $k' \in [2, 8]$ sur un exemple de 6 blocs 3D faiblement séparés entre eux (figure (a)). La valeur optimale est atteinte pour $k' = 6$ et les classes sont représentées sur la figure 3.25 (a) par des symboles différents.

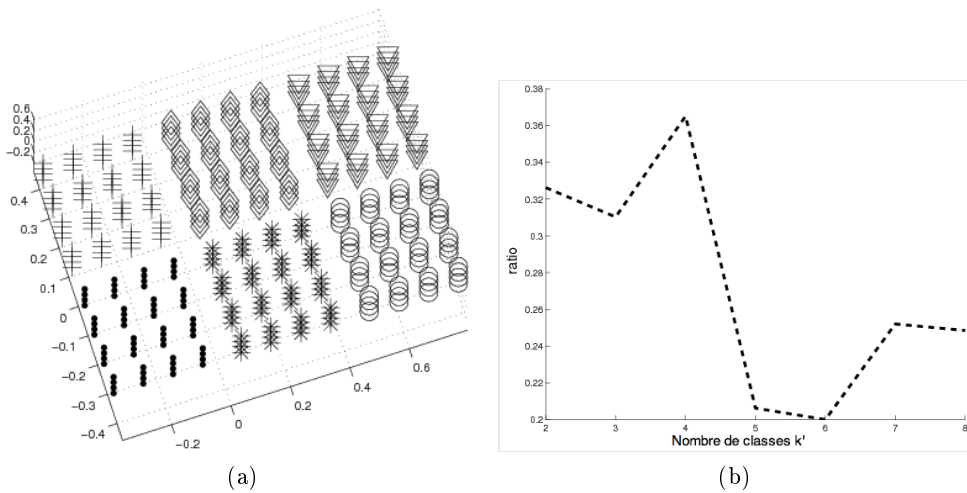


FIGURE 3.25 – Principe de l'heuristique pour le choix du nombre de classes \hat{k} (3.29).

4.3 Application de la classification spectrale pour la recherche de similarités

Dans cette tâche, les isométries correspondantes à des paires d'*entités de référence* sont divisées en six sous-espaces des isométries en fonction de la nature de ces isométries. Dans chaque sous-espace, une classification est requise pour regrouper les isométries proches afin d'identifier les régions similaires dans le modèle. A priori, nous n'avons pas connaissances sur le nombre de classes dans chaque sous-espace. Alors, nous considérons des méthodes de classification non supervisée qui sont capable de déterminer automatiquement le nombre de classes. Mitra et al. [Mitra 2006] propose d'utiliser la méthode Mean Shift, mais l'estimation difficile de la taille de la fenêtre rend les résultats instables. Dans notre travail, nous proposons d'utiliser la méthode de classification Spectrale [Mouysset 2011] qui est plus facile à utiliser et plus efficace que Mean Shift. En effet, il nous suffit de définir la métrique des isométries dans leur espace et de déterminer le nombre maximum de classes.

La figure 3.26 illustre l'efficacité de la méthode Spectrale par rapport à la méthode Mean Shift. En fait, alors que la figure 3.26a présente un modèle symétrique, la figure 3.26b présente le résultat de classification de la classe symétrique par les deux méthodes. Les lignes bleues lient les points de mêmes signatures sur la surface. Idéalement, ces lignes doivent être parallèles. En utilisant la même métrique et sans définir d'autres paramètres, la méthode Spectrale donne un meilleur résultat que la méthode Mean Shift.

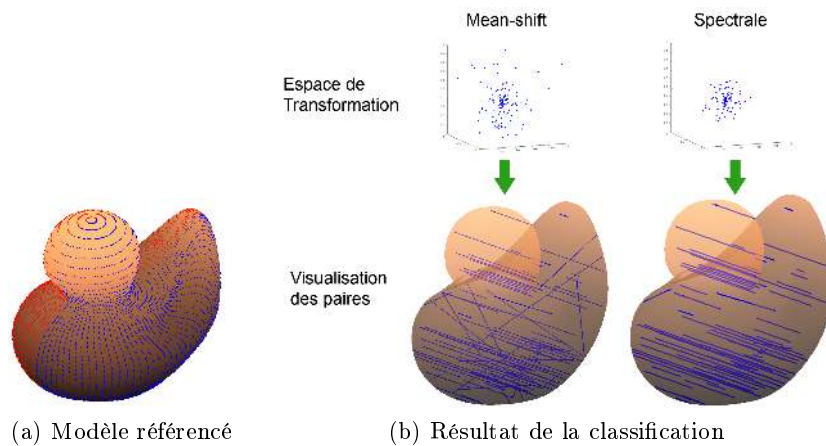


FIGURE 3.26 – Comparaison d'une même classe définie par la méthode Mean-shift et la classification Spectrale.

5 Validation

Après l'extraction des isométries, nous avons un ensemble de paires d'entités avec leurs isométries correspondantes. Ces paires d'entités correspondent à l'un des cas suivants :

- Les entités représentent deux faces topologiquement similaires, c'est-à-dire, dont les sommets sont similaires à une isométrie près. Cependant, ces faces sont aussi définies par les surfaces NURBS sous-jacentes. Pour tester si ces deux faces sont similaires, il faut valider la similarité entre les deux surfaces sous-jacentes lorsque l'on applique l'isométrie correspondante.
- Les entités représentent deux faces dont les sommets ne sont pas mis en correspondance mais il existe probablement une isométrie entre elles. Il faut donc valider cette isométrie entre ces deux faces.
- Les entités représentent deux points de même signature, ils sont soit sur la même face soit sur des faces différentes. Il n'est pas certain que deux régions de surfaces différentes qui les contiennent soient similaires à l'isométrie estimée à partir des repères de ces deux points. Il faut valider la similarité entre ces deux régions avec l'application de l'isométrie correspondante.

De plus, il existe plusieurs classes d'isométries, chaque classe a ses propres caractéristiques. Alors, différentes méthodes de validation sont requises pour satisfaire tous les cas possibles. Dans ce travail, nous proposons quatre méthodes de validation différentes. Chaque méthode est appropriée pour les cas précédents de paires identiques et leurs isométries correspondantes.

5.1 Validation itérative

Dans cette méthode de validation, les faces d'un modèle BRep sont considérées comme des ensembles de points espacés régulièrement. La grille des points paramétriques générée par l'échantillonnage de distance relative uniforme (voir section 2.4.2) nous donne une facilité pour vérifier la similarité des deux régions autour d'une paire de points (p^i, p^j) similaire à l'isométrie T^{ij} . Les régions similaires autour de ces deux points sont identifiées par un processus itératif. En fait, il existe 8 points voisins autour d'un point donné (figure 3.27). Pour assurer la cohérence des voisinages de deux points, les voisins d'un point sont ordonnés suivant le repère local de ce point. Alors, les 8 voisins de p^i sont transformés par T^{ij} et sont comparés aux 8 voisins de p^j . Si la moyenne des distances euclidiennes entre les points transformés et les voisins de p^j est inférieur à un seuil σ^{Iter} , les voisins de p^i seront dits similaires aux voisins de p^j et seront ajoutés au résultat. Ce processus se répète pour chaque voisin de p^i jusqu'à ce que la condition ne soit plus satisfaite. Dans ce cas, les deux régions limitées par les points satisfaisant la condition sont deux régions similaires à l'isométrie T^{ij} .

5.2 Validation par pondération

Cette validation est appropriée pour les paires (F^i, F^j) de faces dont les sommets sont mis en correspondance l'un avec l'autre. Elle partage l'idée principale de l'algorithme d'échantillonnage par pondération (voir section 2.4.3) : les triplets de sommets correspondants à deux faces sont utilisés pour générer des échantillons qui serviront à tester l'isométrie T^{ij} correspondante. Comme pour la validation itéra-



FIGURE 3.27 – Le voisinage d'un point dans la grille d'échantillons.

tive, les échantillons de la face F^i sont transformés et leurs distances euclidiennes aux échantillons correspondantes de la face F^j sont estimés. La validation est bonne si la moyenne des distances ne dépasse pas un seuil σ^{Pond} prédéfini. Pour les faces ayant plus de trois sommets, leurs sommets sont triangulés par la *Triangulation de Delaunay contrainte* [Paul Chew 1989] avant de tester ces faces par l'isométrie correspondante.

Cette méthode de validation est rapide pour les paires de faces ayant la correspondance entre sommets. Cependant, elle exige que les deux faces à tester aient la même paramétrisation.

5.3 Validation basée histogramme

Dans [Osada 2002], Osada introduit la notion de **distribution de formes** pour mesurer la ressemblance entre des maillages 3D. À partir d'un ensemble de points, une *fonction de forme* est utilisée pour mesurer la distribution géométrique de ces points. La distribution des points d'un modèle correspond à un histogramme de B classes. Osada propose cinq fonctions de forme qui sont faciles à évaluer et qui produisent des distributions invariantes aux transformations rigides (rotation, translation et réflexion). Parmi ces fonctions, nous nous intéressons à la fonction $D1$ qui mesure la distance entre un point fixe et les points sur la surface du modèle. Nous utilisons la distribution de forme pour mesurer la ressemblance entre deux faces du modèle BRep-NURBS en redéfinissant la fonction $D1$.

La fonction de forme F^{Hist} mesure la distance euclidienne (norme L^2) d'un point de la face F à une *référence géométrique* qui est généralement l'ensemble des points fixes de l'isométrie T . Comme il existe trois différents ensembles des points fixes : un seul point fixe, une droite de points fixes et un plan de points fixes (voir section 3.1.2), cette méthode de validation n'est applicable qu'à l'isométrie représentant la rotation, la symétrie ou la symétrie suivie de la rotation. Alors, F^{Hist} mesure la distance orthogonale d'un point de la face par rapport à une droite, à un plan ou à un point.

Pour estimer l'histogramme $H(F, T)$ d'une face F associée à une isométrie T , nous définissons K un certain nombre de classes de l'histogramme. L'estimation de l'histogramme consiste en trois étapes :

- Les échantillons de la face F sont générés par l'algorithme de distance relative uniforme (voir section 2.4.2).

- La fonction F^{Hist} est utilisée pour évaluer les distances entre ces échantillons et la référence en fonction de la nature de l'isométrie T .
- Ces distances sont normalisées par la diagonale de la boîte englobante du modèle et sont représentées par un histogramme.

En pratique, l'algorithme d'échantillonnage de distance uniforme génère des points espacés régulièrement sur la face. Cela est nécessaire pour l'approche par histogramme [Osada 2002].

Étant donnée une paire de faces (F^i, F^j) et une isométrie T^{ij} , pour vérifier si F^i et F^j sont similaires à l'isométrie T^{ij} , les histogrammes $H^i = H(F^i, T^{ij})$ et $H^j = H(F^j, T^{ij})$ sont estimés. Nous utilisons la métrique de corrélation entre deux histogrammes pour mesurer la similarité entre deux faces :

$$D^{Hist}(F^i, F^j) = \|H^i - H^j\| = \frac{\sum_{k=1}^B (H^i(k) - \bar{H}^i)(H^j(k) - \bar{H}^j)}{\sqrt{\sum_{k=1}^B (H^i(k) - \bar{H}^i)^2 \sum_{k=1}^B (H^j(k) - \bar{H}^j)^2}}. \quad (3.30)$$

où $\bar{H}^k = \frac{1}{B} \sum_{l=1}^B H^k(l)$. Étant donné un seuil $\sigma^{Hist} \in [0, 1]$, les deux faces F^i et F^j sont dites similaires à l'isométrie T^{ij} si et seulement si :

$$D^{Hist}(F^i, F^j) \geq \sigma^{Hist}. \quad (3.31)$$

5.4 Validation basée distance de Hausdorff

Étant donnés deux ensembles finis de points $A = \{a_0, a_1, \dots, a_{n-1}\}$ et $B = \{b_0, b_1, \dots, b_{m-1}\}$. La distance de Hausdorff est généralement utilisée comme une mesure de similarité entre ces deux ensembles de points [Huttenlocher 1993, Jesorsky 2001]. Elle est définie par l'équation :

$$D^{Haus}(A, B) = \max(h(A, B), h(B, A)). \quad (3.32)$$

où

$$\begin{cases} h(A, B) = \max_{a \in A} \min_{b \in B} \|a - b\|. \\ h(B, A) = \max_{b \in B} \min_{a \in A} \|b - a\|. \end{cases}$$

La fonction $h(A, B)$ estime la *distance de Hausdorff directe* de A à B et respectivement $h(B, A)$ de B à A . $h(A, B)$ estime la distance minimale du point $a \in A$ le plus éloigné de l'ensemble B . De même, $h(B, A)$ estime la distance minimale du point $b \in B$ le plus éloigné de l'ensemble A . La distance de Hausdorff $D^{Haus}(A, B)$ mesure la similarité entre ces deux ensembles de points en prenant le maximum entre $h(A, B)$ et $h(B, A)$. Par conséquent, dans la comparaison de deux ensembles de points, la distance de Hausdorff n'exige pas de mettre en correspondance des points d'un ensemble à l'autre [Huttenlocher 1993].

Étant donné une paire de faces (F^i, F^j) et une isométrie T^{ij} , pour vérifier si ces deux faces sont similaires à cette isométrie près, les échantillons sont prélevés de ces deux faces par l'algorithme d'échantillonnage de distance relative uniforme. La distance minimum utilisée dans l'algorithme est appliquée pour toutes ces faces.

Cela assure que la distribution des points sur les surfaces paramétrées différemment soit similaire. Supposons que E^i soit l'ensemble des échantillons de la face F^i et E^j celui de F^j . Nous appelons E^i l'ensemble « référence » et E^j l'ensemble « image ». L'ensemble « référence » est transformé par l'isométrie $T^{ij} : E^{iT} = T^{ij}(E^i)$. En théorie, les points transformés devraient correspondre à l'ensemble « image ». Alors, la distance Hausdorff $D^{Haus}(E^{iT}, E^j)$ est estimée et est ensuite comparée à un seuil prédéfini σ^{Haus} , les deux faces F^i et F^j sont dites similaires à l'isométrie T^{ij} si et seulement si :

$$D^{Haus}(E^{iT}, E^j) \leq \sigma^{Haus}. \quad (3.33)$$

Cette méthode de validation est appropriée à tout type de paires de faces et à toutes classes d'isométries, puisqu'il n'y a aucune contrainte sur la génération des points de deux faces à tester.

6 Extension

Le processus de validation itérative implique une extension des régions de résultat à partir d'une paire de points. Comme deux faces adjacentes d'un modèle BRep partagent un même bord, nous pouvons identifier, pour un point à la frontière d'une face, ses points voisins dans l'autre face adjacente. Ainsi, cette méthode de validation peut étendre des régions similaires à partir des faces contenant les paires de points vers les faces adjacentes. Pourtant, lors de l'extension, il faut assurer que les deux régions du résultat ne se couvrent pas.

Les autres méthodes de validation appliquées aux paires de faces, impliquent une extension aux faces adjacentes partageant une même isométrie. En effet, la classification des isométries regroupe les paires de faces de même caractéristiques. De plus, lors de l'extraction des isométries, il y a des paires de faces dont les boîtes englobantes de deux faces sont similaires mais qui ne sont pas mises en correspondance car le nombre de sommets des faces est inférieur à 3 ou il existe une tolérance dans le calcul des sommets. Cependant, il peut exister des isométries entre eux. Alors, pour chaque groupe d'isométries, une extension est effectuée en validant l'isométrie représentant ce groupe sur ses paires de faces.

7 Expérimentations

Dans ce paragraphe, nous introduisons les résultats des expérimentations de notre travail. Les expérimentations sont effectuées par deux méthodes d'extraction des isométries : basée topologie et basée géométrie. Toutes les expérimentations sont exécutées sur un PC doté un CPU Core(TM) i7 à 1.73GHz et une mémoire à 8MB.

Tout d'abord, l'utilisation des isométries dans notre travail nous apporte deux grands avantages à l'algorithme de détection de similarités :

- L'identification des isométries en transformations canoniques.
- La caractérisation des transformations canoniques fournit de bonnes métriques pour la classification des isométries.

La figure 3.28 illustre les isométries indirectes détectées par notre algorithme dans un modèle synthétisé. Dans ces images, les points rouges et bleus couvrent les parties similaires du modèle.

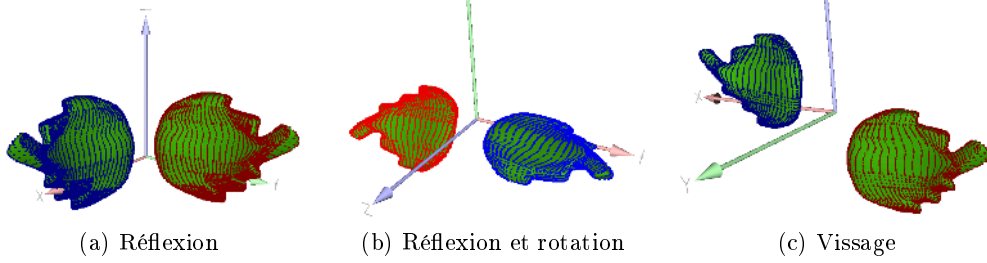


FIGURE 3.28 – Isométries indirectes détectées.

Dans le travail de Mitra et al. [Mitra 2006], ils caractérisent chaque paire de points dans l'espace de transformations par un vecteur $T = (r_x, r_y, r_z, t_x, t_y, t_z) \in \mathbb{R}^6$, où (r_x, r_y, r_z) et (t_x, t_y, t_z) sont les angles d'Euler et la translation extraits de la matrice de transformation entre deux points. Ils définissent la métrique suivante :

$$\|T\|^2 = \alpha(r_x^2 + r_y^2 + r_z^2) + \beta(t_x^2 + t_y^2 + t_z^2)$$

où α et β sont des poids pour régler le biais entre la rotation et la translation. Nous appliquons cette métrique et la notre (section 3.2.2) pour classifier les paires de points. La figure 3.29 illustre un modèle synthétisé où il existe deux réflexions entre les feuilles. Il y a donc deux classes de réflexions différentes. Dans ces deux images, les lignes qui lient les points de mêmes courbures sont colorées rouge ou bleu suivant la classe dans l'espace de transformations. La figure 3.29a présente le résultat avec l'approche de Mitra, nous constatons qu'il y a des points qui sont mal classés. Par contre, dans la figure 3.29b, le résultat de notre approche est meilleur.

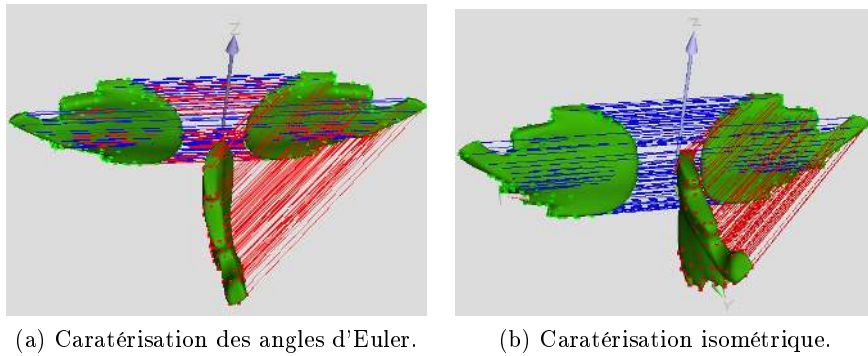


FIGURE 3.29 – Résultats de la classification.

La figure 3.30 illustre les résultats de la détection de similarités avec l'extraction basée géométrie. La figure 3.30b illustre un canard qui est modélisé par une seule

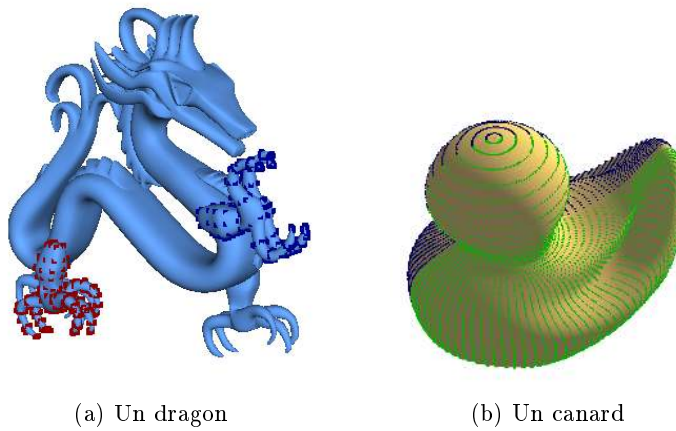
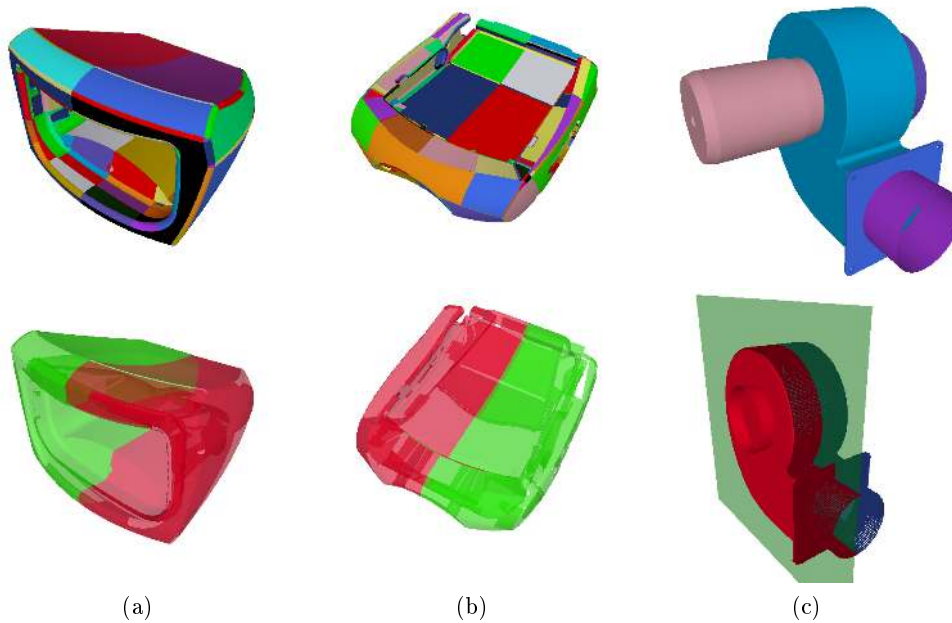


FIGURE 3.30 – Résultat de l'extraction basée géométrie.

face, la réflexion entre deux parties du modèle est bien détectée. De même, le dragon sur la figure 3.30a est composé de plusieurs *faces*. La translation suivie par la rotation entre les deux pattes est bien détectée.

FIGURE 3.31 – Résultats de la détection dans des composantes de CAO.¹

La figure 3.31 illustre les résultats de la détection de réflexion dans des composantes de CAO. Les images de dessus sont des modèles en entrée. Chaque modèle est composé de plusieurs faces qui sont colorées différemment. Les images au dessous

1. Images de Courtoisie PSA.

présentent les résultats de la détection des régions symétriques dans ces modèles. Alors que les autres modèles sont symétriques, le modèle de la figure 3.31c présente une isométrie partielle. Dans l'algorithme de détection, nous appliquons l'extraction des isométries basée topologie. Nous constatons que les régions symétriques globales ou partielles sont bien détectées par l'algorithme.

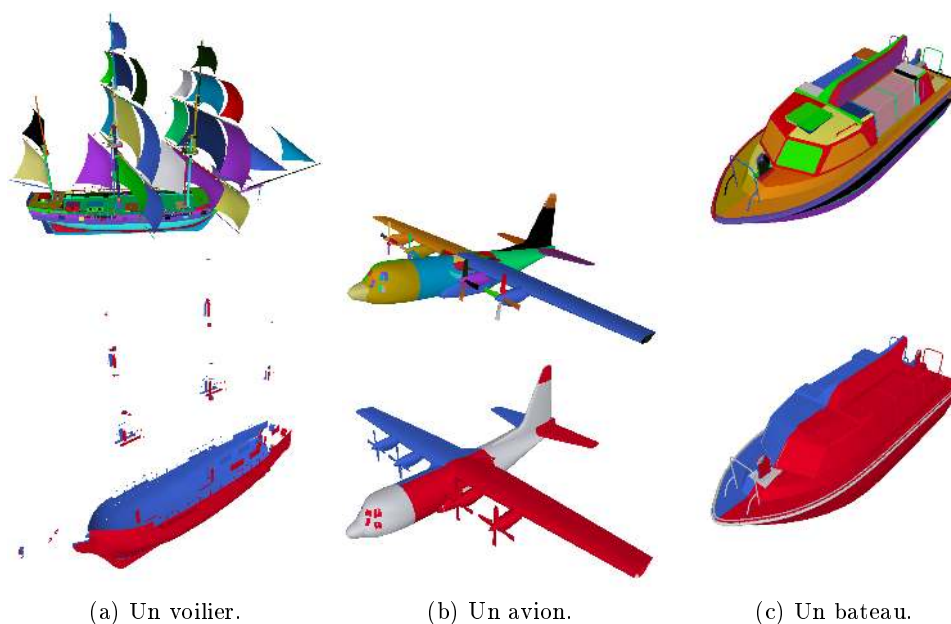


FIGURE 3.32 – Résultats de la détection dans des modèles complexes.

Modèle	(1)	(2)	(3)	(4)	(5)	(6)
Avion	434	517	0,70845	0,02407	0,436407	1,16893
Bateau	622	28	0,74372	0,02167	0,95289	1,71828
Voilier	2538	1742	38,44036	0,2109	3,34267	41,99393

TABLE 3.1 – Temps d'exécution de chaque étape de l'algorithme mesuré en seconde. (1) Nombre de faces, (2) Nombre de paires de faces, (3) Temps d'extraction, (4) Temps de classification, (5) Temps d'extension, (6) Temps total.

La figure 3.32 représente des modèles plus complexes dont les temps d'exécution (en secondes) de chaque étape de notre algorithme de détection sont précisés sur le tableau 3.1. Nous constatons que plus le modèle présente de faces, plus les étapes de l'algorithme (en particulier l'étape d'extraction des isométries) sont coûteuses. De plus, les modèles présentant des faces complexes, par exemple le voilier à la figure 3.32a, prennent davantage de temps pour la mise en correspondance des sommets.

8 Conclusion

Dans ce chapitre, nous avons présenté notre algorithme de détection de similarités dans des modèles BRep-NURBS. Notre algorithme s'inspire du principe de *Procédé du style Hough* qui a généré plusieurs travaux récents. Il est capable d'identifier non seulement les régions similaires d'un modèle mais aussi la transformation qui lie ces régions. Comme les modèles BRep-NURBS sont composés de faces, le problème de la détection des similarités dans ces modèles est réduit à la détection des similarités entre les faces : c'est l'idée principale de l'algorithme. Nous considérons l'utilisation des *loops* extérieures des faces dans la mise en correspondance comme un critère nécessaire pour trouver les faces similaires. Puis, la similarité de deux faces est assurée par une validation qui est considérée comme un critère suffisant. Néanmoins, les topologies des modèles BRep-NURBS sont très variées, ces modèles peuvent être composés d'une ou plusieurs faces. Pour les modèles ayant une seule face où la détection de similarités par face échoue, l'approche de points discrets est privilégiée. Ici, les informations de la géométrie différentielle sont utilisées pour la mise en correspondance. Alors, notre algorithme est divisé en deux processus de traitement différents mais très proches l'un à l'autre : l'un utilise les informations topologiques, l'autre considère les informations géométriques différentielles.

Le facteur principal qui conduit au succès de ce travail est l'utilisation des isométries pour caractériser les similarités. En se basant sur la nature des points fixes, les similarités sont classifiées en plusieurs classes d'isométries qui sont composées de transformations canoniques : la translation, la rotation et la réflexion. Cette classification est très avantageuse, puisqu'elle réduit la complexité de l'espace des isométries lors de la classification des isométries. En fait, un espace qui contient tout type d'isométries est divisé en six sous-espaces différents dédiés à chaque type d'isométries. En outre, l'utilisation de la méthode de classification Spectrale est aussi un apport dans notre travail. Cette méthode est capable de partitionner les groupes d'isométries de façon non supervisée, il nous suffit de définir la métrique des isométries dans leur espace.

Malgré les résultats significatifs, notre algorithme admet quelques limitations :

- Pour l'extraction d'isométries basée topologie, il y a deux problèmes sur la topologie des faces. Premièrement, les *loops* circulaires (figure 3.33a) ad-

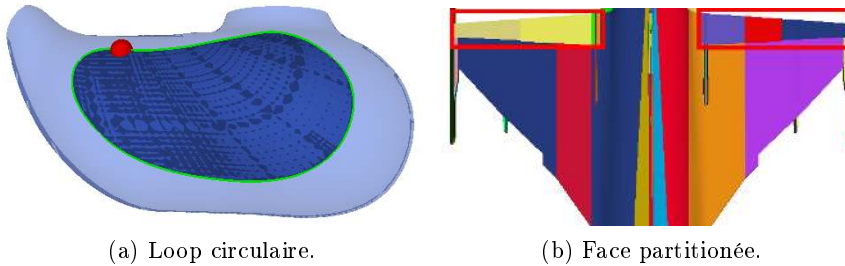


FIGURE 3.33 – Cas dégénérés.

mettent un seul sommet ou plusieurs sommets positionnés arbitrairement. Les mêmes *loops* circulaires peuvent avoir différentes topologies. Ces faces ne peuvent pas être mises en correspondance. Deuxièmement, les faces sont probablement décomposées en plusieurs faces adjacentes (figure 3.33b). Donc, il existe peut-être le cas où une face est similaire à un ensemble de faces. Notre algorithme ne considère pas ces genres de cas.

- Pour l'extraction d'isométries basée géométrie, notre algorithme ne considère pas les points ombiliques dont les directions principales sont mal définies. Alors, notre algorithme n'arrive pas à détecter les similarités dans les faces dont la forme est régulière telle que la forme sphérique et la forme plane.

Application de la détection d'isométries partielles

Sommaire

1	Alignement des modèles 3D	83
1.1	Introduction	83
1.2	Approche	87
1.3	Application à l'indexation des modèles BRep-NURBS	91
1.4	Conclusion	94
2	Compression des modèles	94
2.1	Introduction	94
2.2	Travaux précédents	95
2.3	Approche proposée	96
2.4	Conclusion	98

1 Alignement des modèles 3D

Dans cette section, nous proposons une application de la détection de l'isométrie partielle à l'alignement de modèles BRep-NURBS. Après une introduction présentant les travaux sur ce domaine, notre algorithme d'alignement est ensuite défini et appliqué à l'indexation d'objets 3D.

1.1 Introduction

Grâce au développement des applications en modélisation géométrique dans plusieurs domaines scientifiques, les modèles 3D sont de plus en plus nombreux et sont disponibles à travers plusieurs modalités : des répertoires locaux au sein des organisations ou des dépôts publics au sein des forums professionnels. D'après [Iyer 2005], 75% des activités de conception de produits reproduisent ou adaptent des modèles existants pour générer de nouveaux modèles. La réutilisation est donc une méthode classique pour augmenter l'efficacité du processus de création des produits. De ce fait, les répertoires des objets 3D exigent des techniques efficaces supportant la comparaison ainsi que l'indexation des objets 3D pour faciliter l'acquisition et la manipulation de ces objets. Théoriquement, l'indexation des objets 3D consiste en trois étapes consécutives [Tedjokusumo 2006] : (1) la normalisation, (2) l'extraction des caractéristiques et (3) la comparaison. La première étape est très importante

puisqu'elle affecte fortement le résultat final. En effet, les modèles 3D sont arbitrairement générés en terme d'orientation, de position et de mise à l'échelle. Sans la normalisation, la comparaison des modèles 3D est loin d'être fiable. En d'autres termes, étant donné un modèle 3D, la normalisation consiste à l'aligner suivant un *repère de référence* orthogonal canonique et à le positionner au *centre* de ce repère avant de le mettre à l'échelle. En conséquence, l'alignement est une tâche primordiale et est considéré comme un prétraitement nécessaire à toutes les méthodes d'indexation.

Pour les maillages, de nombreuses techniques ont été proposées et peuvent être classifiées en trois approches : *ACP*, *Symétrie* et *Aire de Projection*. De façon intuitive, l'Analyse en Composantes Principales (ACP) est une technique classique pour aligner les modèles 3D, en particulier les modèles maillés. Vranić [Vranić 2004] aligne les modèles maillés suivant leurs trois axes principaux. L'ACP est raisonnablement simple et efficace, elle utilise les informations statistiques des coordonnées du nuage de points (sommets du maillage) d'un modèle 3D pour identifier les trois directions principales orientées vers les régions dominantes. Ces trois directions sont utilisées pour définir le repère de référence. Cependant, cette technique est très sensible à la distribution des points. Effectivement, comme le montre la figure 4.1, l'ACP est appliquée sur deux modèles BRep-NURBS de notre répertoire. Mais, même avec un échantillonnage de distance relativement uniforme, les axes des repères de référence trouvés par l'ACP de l'avion et de la tasse ont une inclinaison différente de celle attendue. Ces repères donnés par l'ACP ne sont donc pas toujours de bons candidats pour l'alignement des modèles.

D'autres techniques basées sur l'ACP essaient de résoudre le problème de sensibilité de la distribution des points comme l'ACP Continues (CPCA) et l'ACP

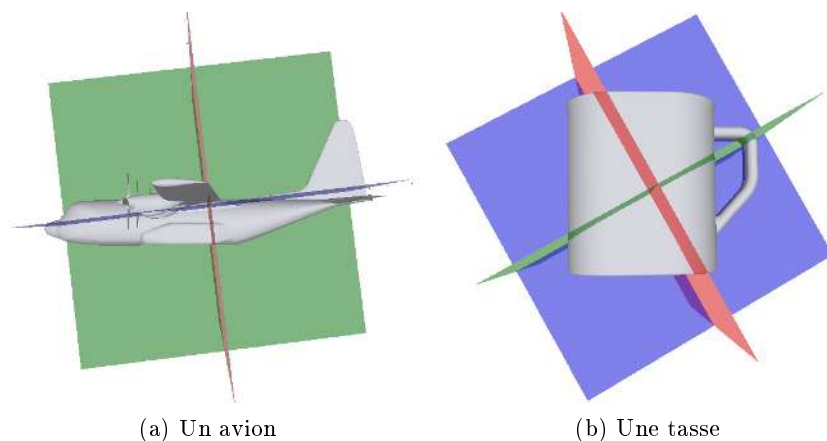


FIGURE 4.1 – Alignement basé sur l'ACP. Les repères canoniques sont définis par trois plans orthogonaux (rouge, vert, bleu), correspondant aux axes principaux dans l'ordre décroissant. Pour l'avion, le plan bleu s'incline à cause de la présence de la queue 4.1a. Pour la tasse, l'anse empêche l'orientation horizontale du plan vert.

Normalisée (NPCA) [Papadakis 2007]. La figure 4.2 présente les résultats de l'alignement de 4 objets 3D. Nous constatons que les objets des colonnes (a),(b) sont mieux alignés avec la NPCA, les objets des colonnes (c),(d) sont mieux avec la CPCA. Alors, ces techniques ne fournissent pas une résolution générale et elles restent peu robustes pour certaines formes d'objets 3D.

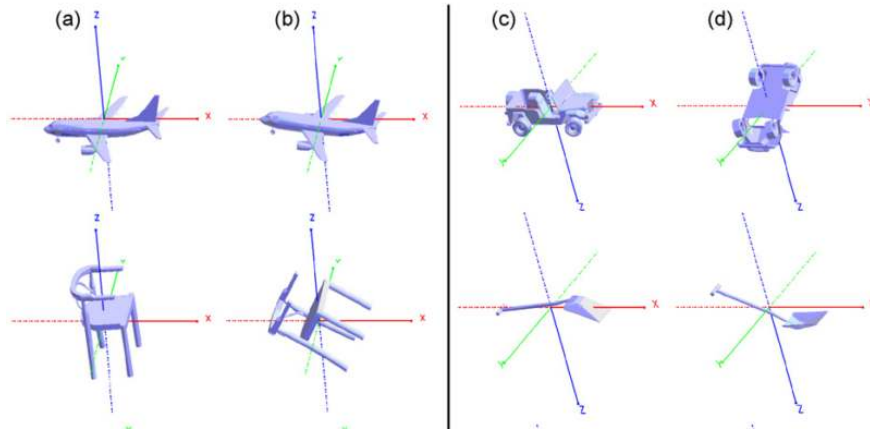


FIGURE 4.2 – Alignement avec NPCA (colonnes a, d) et CPCA (colonnes b, c).

D'après Ferguson et al. [Ferguson 2000], les symétries 2D verticales conduisent à une orientation cohérente avec celle de la perception humaine. La symétrie globale d'un objet peut donc naturellement contribuer à la définition du repère de référence. De plus, Minovic et al. [Minovic 1993] montrent que le plan de symétrie d'un objet 3D est toujours perpendiculaire aux axes principaux. L'information des symétries d'un objet 3D peut donc nous aider à identifier le repère de référence pour l'alignement. Heureusement, la similarité existante sur un objet 3D est un phénomène fréquent [Hel-Or 2010, Mitra 2012]. De nombreux objets sont composés de parties similaires à une rotation, à une translation ou à une symétrie près. Par conséquent, le repère de référence d'un objet 3D peut être identifié grâce à une information de symétrie partielle ou globale. C'est l'idée principale de la deuxième approche d'alignement.

Podolak et al. [Podolak 2006] proposent une *transformée de Symétrie Réflective Planaire* pour estimer tous les plans de symétrie possibles pour un objet donné et alignent suivant les plans de symétrie. Cependant, ils ne prennent en compte que des symétries globales de la forme 3D et non les symétries partielles de l'objet. Chaouch et al. [Chaouch 2008] estiment un repère de coordonnées normalisées en analysant le plan de symétrie extrait par l'ACP Continues. Tedjokusumo et al. [Tedjokusumo 2006] développent aussi une technique basée sur un plan de symétrie bilatérale associé à l'ACP en 2D. Ces techniques considèrent seulement des symétries globales des formes 3D ; elles sont donc restreintes aux objets 3D ayant une symétrie globale.

Récemment, une nouvelle approche basée sur l'*Aire de Projection* a été proposée. Lian et al. [Lian 2010] proposent une méthode utilisant l'ACP et une métrique

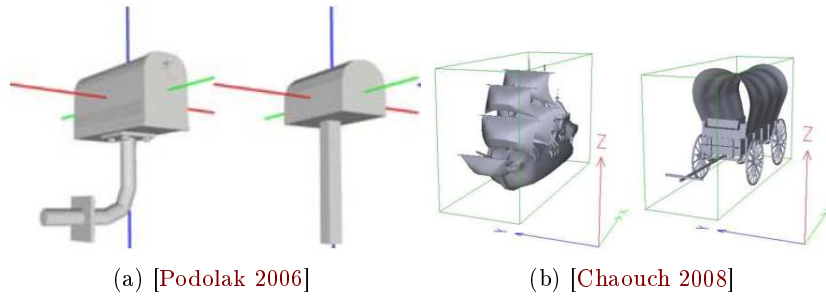


FIGURE 4.3 – Résultats de l'alignement basé symétrie.

de rectification pour identifier tous les repères possibles. Le repère final est celui qui minimise les aires de projection des silhouettes de l'objet 3D. Napoléon et al. [Napoléon 2010] définissent le critère de l'enveloppe visuelle minimum pour identifier les axes principaux à partir des candidats générés par l'ACP. Contrairement à ces deux méthodes qui se basent sur l'ACP, Johan et al. [Johan 2011] proposent une méthode permettant d'identifier les axes du repère de référence en considérant les aires de projection orthographique de l'objet 3D. Étant donné un repère de référence, tous les objets 3D admettent 6 vues canoniques de la projection orthographique (figure 4.4). Ces vues sont réduites à 3 vues principales en considérant seulement l'ensemble des pixels projetés (sans texture) : la vue avant vs. la vue arrière, la vue gauche vs. la vue droite et la vue dessus vs. la vue dessous.

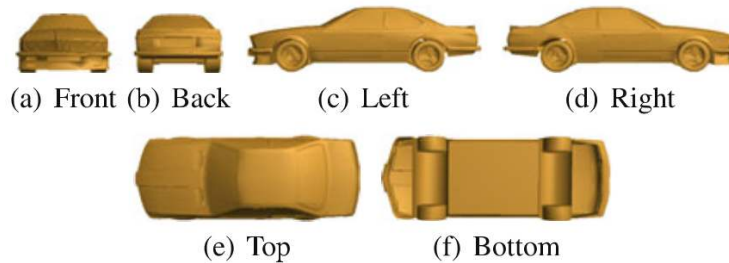


FIGURE 4.4 – Six vues canoniques de la projection orthographique, de gauche à droite et de haut en bas : avant, arrière, gauche, droite, dessus, dessous.

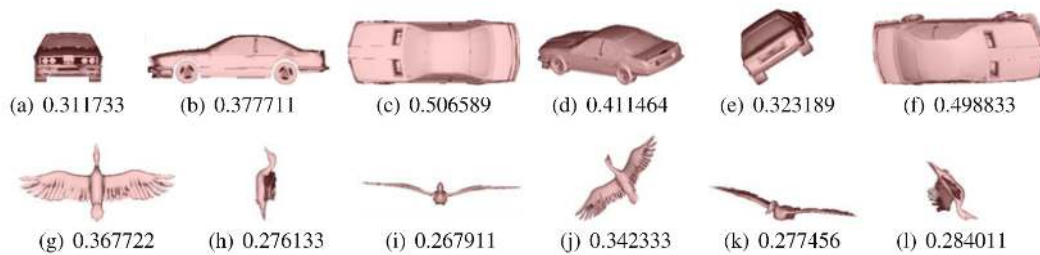


FIGURE 4.5 – Différentes vues de la projection orthographique de deux objets 3D.

Johan et al. constatent que pour la plupart des objets, une des trois vues canoniques admet une aire de projection minimale plus petite que les autres vues arbitraires (figure 4.5). Ils proposent alors un algorithme appelé *Alignement par minimisation de projection* (AMP), cet algorithme est divisé en trois étapes :

1. Estimer de façon *gloutonne* (greedy) toutes les vues possibles autour de l'objet, choisir la vue minimisant l'aire de projection de l'objet pour définir le premier axe.
2. Choisir parmi les vues tournant autour du premier axe la vue minimisant l'aire de projection de l'objet pour le deuxième axe.
3. Dédire le troisième axe par le produit vectoriel.

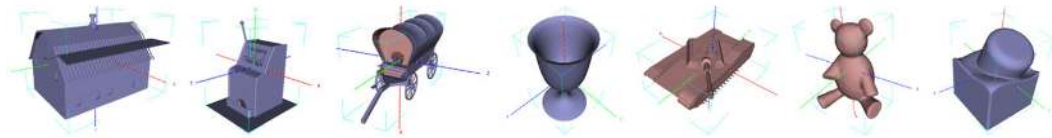


FIGURE 4.6 – Résultats de l'alignement par MPA.

Malgré des résultats visiblement intéressants et convaincants (figure 4.6), l'algorithme proposé ne génère pas toujours des résultats cohérents avec la perception visuelle humaine car il ne prend pas en compte de critères perceptifs. De plus, dans la première étape, l'algorithme doit considérer tous les plans autour de l'objet. C'est une estimation très coûteuse qui affecte la performance de l'algorithme.

Après un tour d'horizon des approches d'alignement, nous constatons que la prise en compte, des symétries globales seules ou des aires de projection seules des modèles 3D, peut-être insuffisante ou coûteuse pour aligner tout type d'objets. De ce fait, nous proposons une méthode d'alignement qui utilise les symétries partielles existant dans des modèles 3D, et les associe aux aires de projection orthographique, pour développer un algorithme d'alignement plus efficace.

1.2 Approche

Nous proposons un algorithme d'alignement des modèles BRep-NURBS intuitif et consistant avec la perception visuelle humaine. Étant donné un modèle BRep-NURBS, nous choisissons le premier axe du repère de référence comme la normale du plan de la symétrie partielle dominante. Ensuite, nous choisissons, parmi les plans orthogonaux au plan de la symétrie, le plan qui minimise l'aire de la projection orthographique du modèle : la normale à ce plan est considérée comme le deuxième axe. Le troisième axe est déduit par produit vectoriel à partir de ces deux axes. L'alignement proposé se divise donc en deux étapes :

1. *La détection de symétries* identifie le plan de la symétrie partielle dominante.
2. *La rectification du repère de référence* estime les deux axes restants du *repère de référence*.

Dans cet algorithme, nous avons une nouvelle notion, la **symétrie partielle dominante**, nous la considérons comme la symétrie dont la boîte englobante de deux régions similaires est la plus grande. De plus, nous considérons seulement les modèles BRep-NURBS qui sont composés de plusieurs faces.

1.2.1 Détection de la réflexion partielle dominante

Cet algorithme est similaire à l'algorithme introduit dans le chapitre 3. Sauf qu'il identifie seulement les régions similaires à une *symétrie planaire* près. Les résultats de la détection des réflexions sont ensuite utilisés pour la projection orthographique du modèle. Pour avoir une bonne projection, il faut que l'algorithme détecte le plus de faces possibles.

Pour une réflexion détectée, il existe peut-être des faces qui ne sont pas mises en correspondance avec une autre face suivant cette réflexion mais elles sont (potentiellement) autosimilaires et donc intersectent le plan de réflexion. Nous prenons aussi en compte ces faces. Cependant, il nous faut valider les parties de ces faces autosimilaires par cette réflexion. Nous adaptons la validation basée distance de Hausdorff ou basée histogramme pour ce problème. Les deux ensembles à valider sont déterminés comme étant respectivement à gauche et à droite du plan de réflexion.

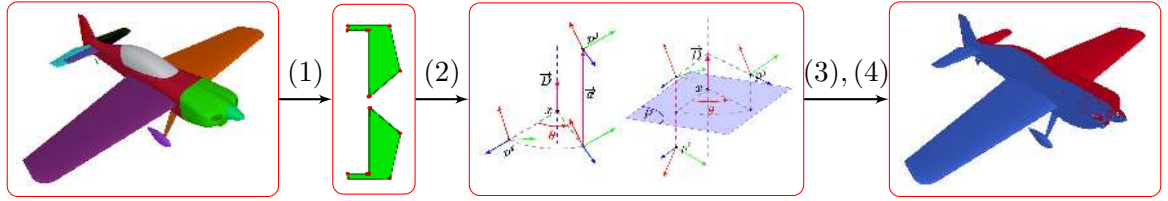


FIGURE 4.7 – Algorithme de détection de symétrie partielle dominante.

Comme illustré dans la figure 4.7, notre algorithme de détection de symétrie partielle dominante consiste en trois étapes :

1. Extraction des symétries et validation.
2. Classification des symétries.
3. Extension vers les faces intersectant le plan de réflexion.
4. Identification de la symétrie partielle dominante.

La symétrie partielle dominante d'un modèle est celle ayant des faces symétriques dont le volume de la boîte englobante est le plus grand. Pour éviter le problème de surdimensionnement, cette boîte englobante est considérée comme l'union des boîtes englobantes des faces symétriques. La figure 4.8 illustre les résultats de la détection de symétrie planaire dans trois modèles BRep-NURBS. Les régions colorées de façon continue représentent des faces similaires, les régions colorées de façon discrète représentent les faces intersectant le plan de réflexion. La figure 4.8c présente la symétrie partielle dominante dans un modèle non symétrique.

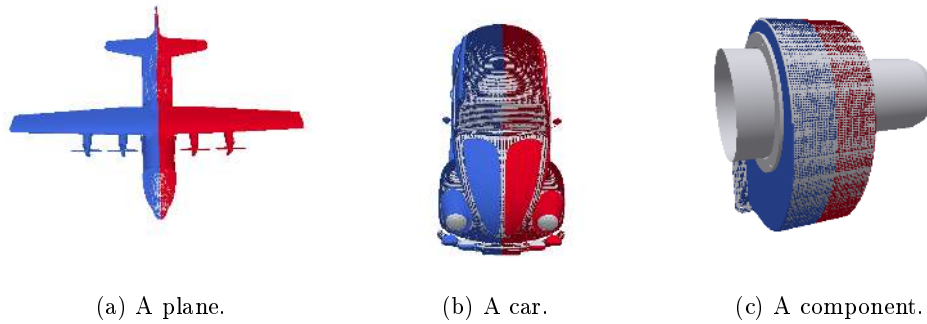


FIGURE 4.8 – Les symétries partielles dominantes détectées.

1.2.2 Rectification du repère de référence

Le premier axe du repère de référence d'alignement est choisi comme la normale du plan de la symétrie partielle dominante. Cet axe est appelé *axe de référence*. Dans ce paragraphe, nous présentons notre algorithme d'extraction des deux axes restants. En fait, il suffit de déterminer un *deuxième axe*, le troisième axe est déduit à partir de l'axe de référence et du deuxième axe car le repère est orthogonal. Nous utilisons une méthode de projection orthographique du modèle sur les plans candidats perpendiculaires au plan de référence. Pour générer ces candidats, à partir d'une orientation arbitraire perpendiculaire à l'axe de référence, les normales aux plans candidats sont déterminées en tournant autour de l'axe de référence. Nous considérons 180 plans candidats consécutifs qui sont espacés d'un angle de 1° . La figure 4.9 présente les projections des modèles sur quelques plans candidats.

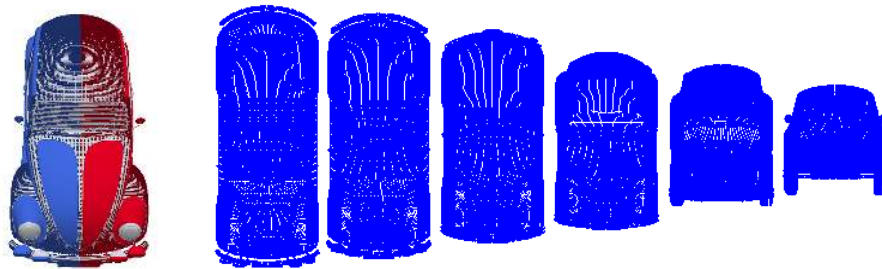


FIGURE 4.9 – Aires de projection du modèle d'une voiture. L'image à gauche représente le deux parties de la symétrie partielle dominante, les images à droite sont des projections du modèle sur les plans de candidat.

Pour chaque plan candidat, l'aire de projection orthographique est estimée par le nombre de pixels de la forme 2D projetée. Le candidat choisi est celui qui contient l'aire de projection la plus petite.

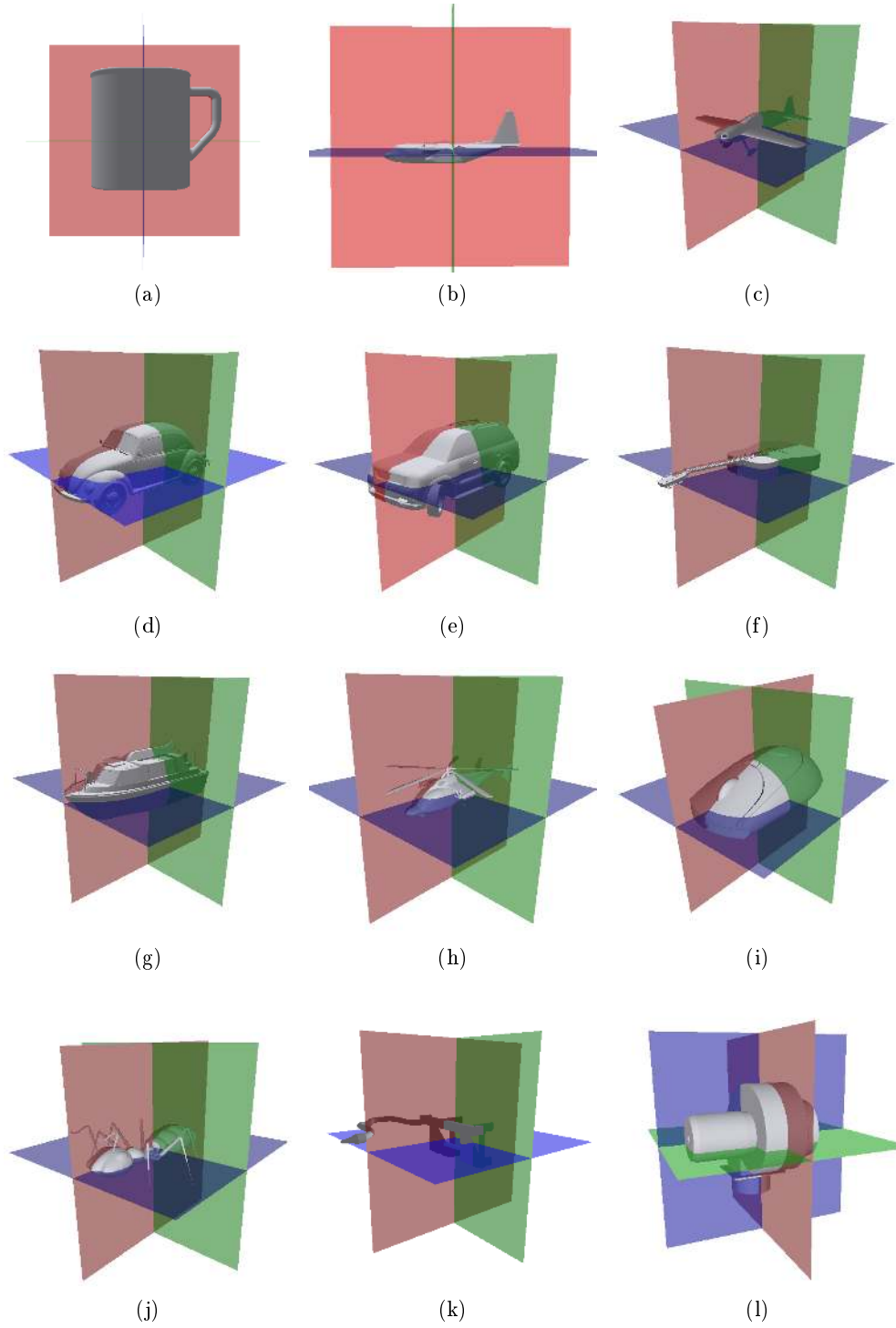


FIGURE 4.10 – Résultats de l'alignement des modèles BRep-NURBS.

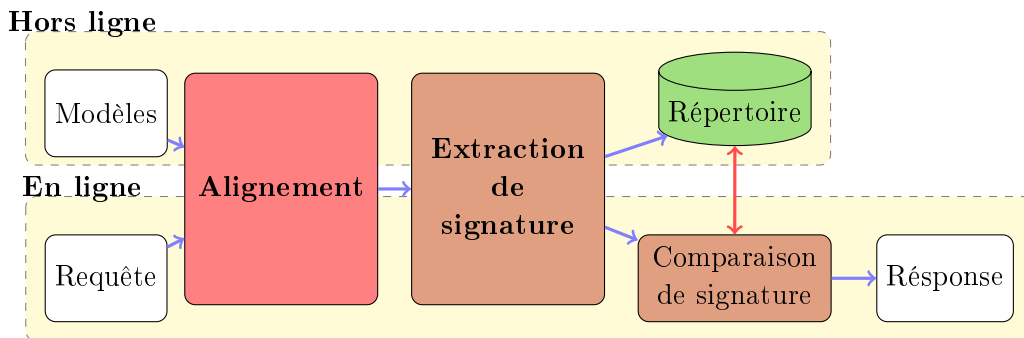
1.2.3 Expérimentations

Dans notre travail, les modèles BRep-NURBS sont ceux du site <http://www.grabcad.com>. Les modèles sont classifiés dans plusieurs catégories telles que *voiture*, *avion*, *bateau*, *instrument*, *composant mécanique*. Les images de la figure 4.10 exposent les résultats d'alignement de nos modèles. Pour chaque image, les trois plans rouge, vert et bleu représentent les trois plans du repère de référence d'alignement. Le plan rouge est le plan de la symétrie partielle dominante du modèle. Le plan vert est le deuxième plan qui minimise l'aire de projection orthographique du modèle.

Comme le montrent ces images, notre technique d'alignement est efficace pour plusieurs types d'objets malgré leurs formes. En particulier, les résultats de deux figures 4.10a et 4.10b montrent que les résultats dérivés de notre technique sont plus aptes que les approches d'ACP. En effet, l'utilisation de la symétrie partielle dominante rend notre algorithme cohérent avec la perception visuelle humaine. Dans la partie suivante, nous appliquons notre algorithme d'alignement à l'indexation des modèles BRep-NURBS.

1.3 Application à l'indexation des modèles BRep-NURBS

Dans ce paragraphe, nous appliquons les résultats dérivés de notre algorithme d'alignement à l'indexation des modèles dans notre base de modèles BRep-NURBS. La structure générale de notre système d'indexation est illustrée comme suit :



Notre système est un système d'indexation de « requête par modèle ». C'est-à-dire, en fournissant une requête sous forme d'un modèle 3D, le système retourne les modèles existants ayant une forme similaire à celle du modèle à l'entrée. Cette similarité est mesurée par un score de similarité entre le modèle requête et un modèle du répertoire. Dans notre système, il y a deux parties qui fonctionnent indépendamment :

- **Hors ligne** : les signatures des modèles existants sont estimées et stockées dans un répertoire.
- **En ligne** : quand l'utilisateur fournit un modèle en entrée, la signature du modèle est estimée et comparée avec les signatures du répertoire pour identifier les modèles similaires.

Dans les deux parties, il existe un processus commun : les modèles 3D sont alignés avant d'estimer leur signatures. Nous appliquons une approche *basée image* pour évaluer la signature des objets 3D. Les contours de la projection du modèle 3D sur les plans du repère de référence sont extraits et convertis en images 2D où les caractéristiques estimées des silhouettes supportent l'indexation et la recherche des modèles [Liu 2010, Eitz 2012]. En effet, les caractéristiques des silhouettes d'un objet reflètent la perception visuelle quand les humains essaient de décrire cet objet suivant des vues spécifiques. Ici, en utilisant les trois plans du repère canonique d'alignement d'un modèle 3D, les trois contours de ce modèle sont extraits par les projections du modèle sur ces plans (voir figure 4.11). Ensuite, un vecteur de caractéristiques pour plan est évalué. Nous adoptons le descripteur de Fourier normalisé associé avec le calcul des distances barycentriques [Zhang 2001] pour évaluer les caractéristiques des contours.

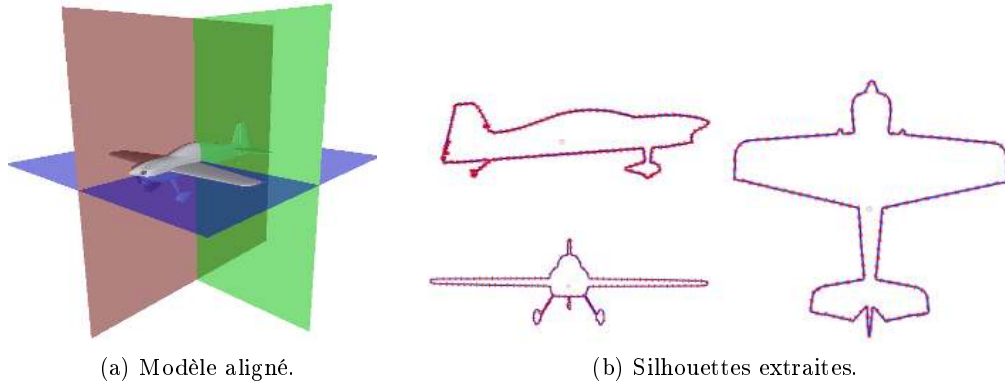


FIGURE 4.11 – Extraction des silhouettes d'un modèle *avion* en se basant sur les plans du repère canonique.

Étant donné un contour défini par une courbe fermée C qui est représentée par une fonction unidimensionnelle $u(t)$, appelée la signature du contour. À tout instant t , il existe un nombre complexe $u(t)$, $0 < t < T$ où T est la période de t , c'est-à-dire $u(t) = u(t + nT)$ avec $n \in \mathbb{N}$. La *transformée de Fourier discrète* pour la fonction $u(t)$ échantillonnée en N points est définie par :

$$FD_k = \sum_{n=0}^{N-1} u(n) e^{-2i\pi k \frac{n}{N}}. \quad (4.1)$$

Les coefficients FD_k , $0 \leq k < N$, sont appelés des descripteurs de Fourier du contour. Effectivement, tous les N descripteurs sont nécessaires à la représentation du contour sauf le premier qui dépend uniquement du point de départ $u(0)$ choisi et ne contribue pas à la description du contour. Alors, le vecteur des descripteurs de Fourier du contour donné devient :

$$V = \left[\frac{|FD_1|}{|FD_0|}, \frac{|FD_2|}{|FD_0|}, \dots, \frac{|FD_{N-1}|}{|FD_0|} \right]. \quad (4.2)$$

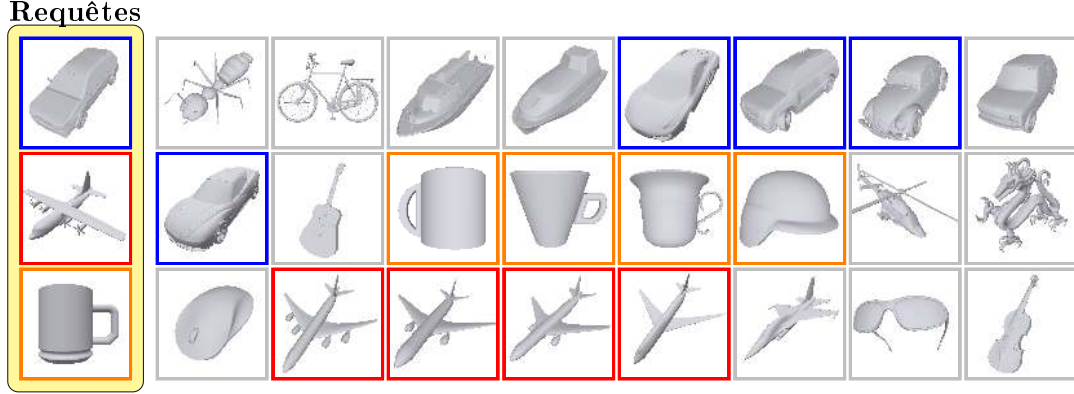


FIGURE 4.12 – Résultats de trois requêtes.

Ces descripteurs sont invariants à la rotation, à la translation et à la mise à l'échelle [Zhang 2001], ils sont donc appropriés à notre algorithme d'alignement. En effet, chaque modèle est caractérisé par trois vecteurs des descripteurs de Fourier normalisés V_1, V_2 et V_3 . Nous définissons ensuite une mesure de similarité entre deux modèles BRep-NURBS comme suit :

$$Sim(M^1, M^2) = \sum_{i=1}^3 D(V_i^1, V_i^2). \quad (4.3)$$

où $D(\bullet, \bullet) \in \mathbb{R}$ est la distance Euclidienne.

Notre base contient environ 100 modèles BRep-NURBS qui sont divisés en plusieurs catégories. Pour évaluer l'efficacité de l'indexation, nous utilisons deux mesures : la *précision* P et le *rappel* R . La précision est définie comme une proportion des solutions pertinentes qui sont trouvées : elle mesure la capacité du système à donner toutes les solutions pertinentes. Le rappel est défini comme une proportion de solutions trouvées qui sont pertinentes : il mesure la capacité du système à refuser les solutions non-pertinentes. De plus, les courbes précision/rappel sont aussi générées pour évaluer l'efficacité de l'indexation 3D en appliquant notre algorithme d'alignement par rapport à l'approche de l'ACP.

La figure 4.12 montrent les résultats de 3 requêtes d'indexation dans notre répertoire. La colonne gauche présente les modèles de requête, la colonne droite présente quelques modèles de notre répertoire. Les modèles de la réponse à droite sont colorés de la même couleur que le modèle de requête à gauche.

Le graphe de la figure 4.13 évalue l'efficacité de notre algorithme d'alignement par rapport à l'approche ACP en appliquant l'indexation des modèles dans notre répertoire. Nous constatons que notre algorithme est plus efficace : sa précision est en moyenne 8% plus grande que celle de l'ACP.

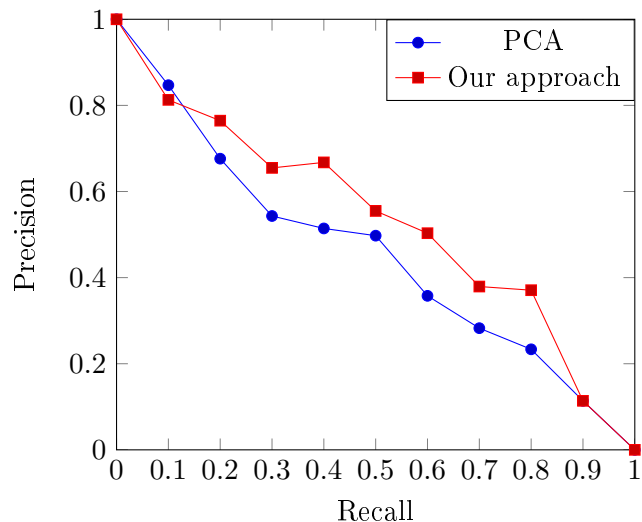


FIGURE 4.13 – Résultats de trois requêtes.

1.4 Conclusion

Dans ce chapitre, nous avons proposé une approche originale pour aligner les modèles BRep-NURBS. Notre algorithme est capable d'aligner des modèles BRep-NURBS de manière plus efficace que des techniques classiques, dépendantes de la distribution des points des surfaces. En effet, notre algorithme se base sur les symétries partielles dominantes des modèles pour identifier le repère canonique d'alignement. Alors que l'utilisation de la symétrie globale était appliquée avec succès sur les maillages dans les travaux précédents, l'utilisation de la symétrie locale associée à la projection orthographique est une approche innovante.

De plus, notre algorithme d'alignement est appliqué à l'indexation des modèles 3D comme une tâche de prétraitement. En utilisant le descripteur de Fourier normalisé des contours 2D, les résultats obtenus dans nos expérimentations s'améliorent significativement.

2 Compression des modèles

2.1 Introduction

Dans cette section, nous présentons une seconde application de la détection de similarités : la compression des modèles BRep-NURBS. La compression de modèles 3D est en particulier utile par la transmission de ces modèles. Comme nous l'avons mentionné, les modèles 3D sont de plus en plus nombreux et sont disponibles à travers plusieurs modalités : des répertoires locaux au sein d'organisations spécifiques ou des dépôts publics. Alors, différentes applications ont été proposées permettant les utilisateurs à consulter les contenus 3D en ligne. Les premières applications sont les *musées virtuels*. Des projets exemplaires de la numérisation d'œuvres artistiques de

musées sont : *the Stanford's Digital Michelangelo* [Levoy 2000] ou *Digital Sculpture* [Miyazaki 2006]. L'accès virtuel aux oeuvres artistiques de musées offrent plusieurs avantages. Tout d'abord, ces applications permettent aux utilisateurs des visites de musées sans déplacement. Ensuite, les modèles virtuels fournissent les accès aux oeuvres qui sont très délicates à exposer au public. Par exemple, *les grottes de Gargas*, un site préhistorique où les peintures sont datées de l'an 2500 avant JC ne peuvent pas accueillir trop de visiteurs afin de préserver l'intégrité du site. Ce site a donc été numérisé pour permettre les visites virtuels. Enfin, les modèles virtuels permettent aux visiteurs de choisir eux-mêmes leurs points de vue préférés.

D'autres applications s'inscrivent dans le contexte du *e-commerce*. La représentation des produits de la plupart des sites web est restée classique, c'est-à-dire, les produits sont présentés par des photos 2D. Cependant, quelques applications en ligne ont récemment proposé aux utilisateurs de visualiser et interagir avec des contenus 3D. Le site <http://p3d.in> propose un outil général pour partager en ligne des modèles 3D, les utilisateurs peuvent visualiser les modèles avant de les télécharger. Toyota¹ fournit la possibilité de configurer les voitures avec des options et de les visualiser en 3D. Dans le domaine de la CAO, les sites <http://grabcad.com> et <http://3dshap.es> proposent des forums de correspondance entre les concepteurs 3D professionnels. Ils peuvent chercher et acheter les modèles 3D existants répondant à leurs besoins. La visualisation des modèles 3D est avantageuse pour ces sites : elle peut, d'une part, donner les détails de modèles aux concepteurs, d'autre part, elle peut protéger la licence privée de ces modèles 3D. De façon classique, les modèles 3D sont pré-visualisés par des vidéo courtes qui ne permettent pas l'interaction 3D avec les utilisateurs. Alors, une autre approche a été proposée, *streaming des aperçus 3D*. Cette approche permet la pré-visualisation du modèle pendant son téléchargement : les parties du modèle correspondant au point de vue couvrant sont envoyées en priorité [Zhao 2013].

Pour ces applications en ligne, le temps de téléchargement des contenus 3D est significatif. La transmission des modèles 3D doit être adaptée aux ressources matérielles des utilisateurs : un terminal léger comme un téléphone portable ou une connexion limitée comme le réseau 3G. Pour cela, les données doivent être les plus compactes possibles, des algorithmes de la compression des contenus 3D sont requis.

2.2 Travaux précédents

La compression des surfaces paramétriques a déjà été abordée. Forsey et al. [Forsey 1988] définissent les modèles B-Spline multi-résolution. À partir d'une surface B-Spline originale, ils proposent un algorithme itératif qui crée les versions plus fines en réduisant le nombre de noeuds et de points de contrôle à chaque itération. Kazinnik et al. [Kazinnik 1997] proposent une décomposition multirésolution par ondelettes pour compresser les surfaces NURBS. Ducci [Ducci 2003] présente une approche de la compression des surfaces NURBS : les vecteurs de noeuds et les points de contrôle sont encodés par des méthodes de prédiction et d'entropie. Les résultats obtenus sont très intéressants, ils appliquent aussi cette approche pour

les surfaces NURBS trimmées. Furukawa et al. [Furukawa 2002] proposent un algorithme utilisant les bords des NURBS et des données supplémentaires. Les courbes des bords d'une surface NURBS sont extraites pour interpoler une surface de type Coons. La différence entre la surface originale et celle interpolée est évaluée pour former deux cartes de distances représentant deux niveaux de précision. La surface NURBS est alors encodée par les courbes de bords et les deux cartes de distances (figure 4.14).

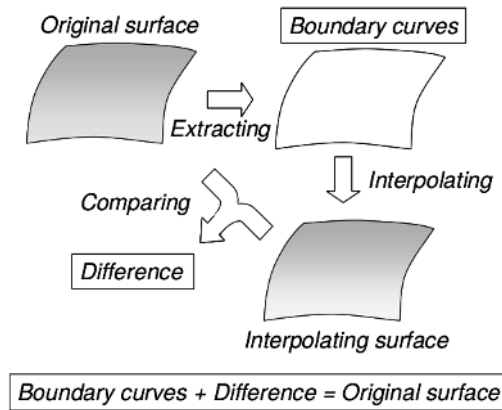


FIGURE 4.14 – L'algorithme de compression de Furukawa et al. [Furukawa 2002].

Nous constatons que les approches précédentes ont utilisé des calculs compliqués. Les approches multi-résolution sont très appropriées pour le *streaming* car elles permettent une visualisation progressive du modèle. Néanmoins, dans le contexte d'un modèle BRep-NURBS avec plusieurs surfaces NURBS (parfois des milliers), le temps de décompression est significatif. Nous proposons alors une approche directe en utilisant les similarités détectées dans un modèle BRep pour qu'il profite des redondances dans ces modèles. De plus, la décompression de ces modèles est efficace car il suffit d'appliquer des transformations affines.

2.3 Approche proposée

Notre algorithme de détection de similarités peut identifier les faces similaires et tout type d'isométries dans un modèle. Alors, pour une paire de faces similaires à une isométrie près, il suffit de préserver une seule face et l'isométrie correspondante pour éviter la répétition. Cela permet de réduire l'espace de stockage des modèles : c'est l'idée principale de la compression des modèles. Dans cette application, nous proposons tout d'abord d'utiliser les réflexions dans les modèles BRep-NURBS pour les compresser. La réflexion choisie est la réflexion dominante dans ce modèle. L'algorithme pour détecter la réflexion partielle dominante a été présenté à la section 1.2.1. La figure 4.15 illustre les modèles que nous utilisons pour tester notre algorithme de compression. Ces modèles sont soit globalement symétriques soit partiellement

1. <http://www.toyota.fr/new-cars>

Modèle	Nombre de faces	Nombre de faces répétées	Taille originale (KB)	Taille compressée (KB)	Ratio de compression
Avion	434	259	98,3	64,2	0,35
Comp1	957	414	212,6	125,4	0,41
Comp2	348	149	79,3	47,9	0,40
Comp3	410	190	92,8	52,6	0,43
Comp4	59	12	20,4	18,4	0,10
Dragon	510	83	114,7	99,0	0,14
Voilier	2538	1064	559,3	401,8	0,28
Voiture	905	258	169,8	119,9	0,29

TABLE 4.1 – Résultats de la compression.

symétriques. Le tableau 4.1 présente les résultats de compression de ces modèles.

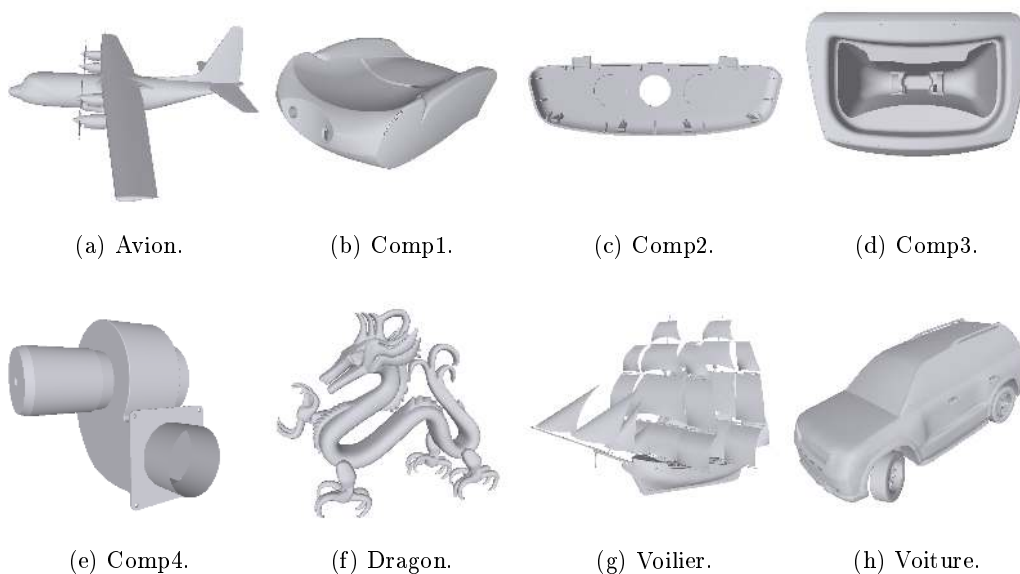


FIGURE 4.15 – Modèles à tester la compression.

Nous constatons que le ratio de compression des modèles globalement symétriques est plus élevé que des modèles partiellement symétriques. Ici, nous considérons seulement la réflexion. Mais il existe d'autres types d'isométries que nous pouvons utiliser pour la compression. Par exemple, le modèle du dragon comporte aussi des translations et des rotations entre les pattes. Donc, l'impact sur le ratio de compression pourrait être bien meilleur en prenant en compte les autres isométries.

2.4 Conclusion

Les premiers résultats de notre approche sont intéressants car ils constituent les premiers pas vers la compression 3D. En effet, en combinant tout type d'isométries, nous nous confrontons à de nouvelles problématiques. Premièrement, une face peut être similaire avec plusieurs faces avec des isométries différentes. Le choix d'une face similaire à enlever dans le modèle appelle à résoudre un problème d'optimisation. Deuxièmement, la dépendance entre les faces similaires, c'est-à-dire une face est similaire à une deuxième face mais elle-même est similaire à une autre, entraîne une complexité dans le stockage. Il faut alors construire une structure arborescente pour enregistrer toutes ces similarités de façon optimale, ce qui crée des dépendances. Ces dépendances seront à prendre en compte lors de la transmission.

Conclusion et Perspectives

Sommaire

1	Conclusion	99
2	Perspectives	101
2.1	Détection de similarités	101
2.2	Applications basées sur les similarités détectées	101

1 Conclusion

Dans la Modélisation Géométrique, les surfaces paramétriques, en particulier les surfaces B-Splines Non-Uniforme Rationnelles (NURBS), constituent un outil performant pour les communautés académiques et industrielles pour la conception et l'analyse des objets 3D. En associant ces surfaces à une Représentation par Bords (BRep) qui est un standard industriel, les modèles BRep basés NURBS (BRep-NURBS) deviennent un standard. Ils sont utilisés largement dans plusieurs domaines tels que l'architecture, l'automobile, l'aéronautique, la mécanique, la biologie. Par conséquent, ces modèles 3D sont de plus en plus nombreux et disponibles à travers plusieurs modalités : des répertoires locaux au sein des organisations spécifiques ou des dépôts publics au sein de forums professionnels. Ils sont en plus très variés au niveau de la fonctionnalité et de la forme. Pour augmenter la productivité dans la création de tels objets virtuels, la réutilisation et l'adaptation des modèles existants est un choix prioritaire. En effet, 75% des activités de conception de produits consistent à reproduire ou adapter des modèles existants pour générer de nouveaux modèles. Cela exige donc des méthodes facilitant le stockage, la recherche et l'exploitation de ces modèles.

La **similarité** est un phénomène fréquent existant dans des objets naturels et synthétiques. De nombreux objets exhibent des parties similaires à une rotation, une translation, une réflexion ou plus généralement à une isométrie près. La similarité est aussi une caractéristique essentielle de la perception humaine. Alors, l'exploitation des similarités dans des modèles BRep-NURBS peut donner des résultats réalistes et utiles satisfaisant des besoins actuels. Cette thèse propose deux contributions principales.

Dans un premier temps, nous avons présenté notre algorithme de détection de similarités dans des modèles BRep-NURBS. Notre algorithme s'inspire du principe du *Procédé de la transformée de Hough* qui a généré plusieurs travaux récents. Il

est capable d'identifier non seulement les régions similaires d'un modèle mais aussi la transformation qui lie ces régions. Comme les modèles BRep-NURBS sont composés de faces, le problème de la détection des similarités dans ces modèles revient à la détection des similarités entre les faces : c'est l'idée principale de l'algorithme. Nous considérons l'utilisation des *loops* extérieures des faces dans la mise en correspondance comme un critère nécessaire pour trouver les faces similaires. Puis, la similarité de deux faces est assurée par une validation qui est considérée comme un critère suffisant. Néanmoins, les topologies des modèles BRep-NURBS sont très variées, ces modèles peuvent être composés d'une ou plusieurs faces. Pour les modèles ayant une seule face où la détection de similarités échoue par la dernière idée, l'approche par points discrets est privilégiée. Ici, les informations de la géométrie différentielle est utilisée pour la mise en correspondance. Alors, notre algorithme est divisé en deux processus de traitement différents mais très proches l'un de l'autre : l'un utilise les informations topologiques, l'autre considère les informations géométriques différentielles.

Le facteur principal qui conduit au succès de ce travail est l'utilisation des isométries pour caractériser les similarités. En se basant sur la nature des points fixes, les similarités sont classifiées en plusieurs classes d'isométries qui sont composées des transformations canoniques : la translation, la rotation et la réflexion. Cette classification est très avantageuse, puisqu'elle réduit la complexité de l'espace des isométries lors de la classification des isométries. En fait, un espace qui contient tout type d'isométries est divisé en six sous-espaces différents dédiés à chaque type d'isométrie. En outre, l'utilisation de la méthode de classification spectrale est aussi un apport dans notre travail. Cette méthode est capable de partitionner les groupes d'isométries de façon non supervisée, il nous suffit de définir la métrique des isométries dans leur espace.

Ensuite, nous avons présenté nos algorithmes d'échantillonnage des faces du modèle BRep-NURBS. Ces algorithmes sont appropriés pour notre algorithme de détection de similarités : les échantillons sont générés à l'intérieur des faces et sont aussi espacés de façon relativement uniforme et la grille régulière des échantillons permet la navigation aux voisins d'un échantillon donné. Puis, nous avons introduit les isométries dans l'espace affine euclidien, un facteur important contribuant au succès de la détection de similarités dans ce travail. En effet, les isométries sont considérées comme des transformations canoniques : rotations, translations, symétries ou leurs compositions. La décomposition et la classification des isométries en transformations canoniques sont réalisées par le support des points fixes qui, à leur tour, fournissent des caractéristiques invariantes des isométries.

Dans un deuxième temps, nous avons proposé une approche originale pour aligner les modèles BRep-NURBS. Notre algorithme est capable d'aligner des modèles BRep-NURBS de manière plus efficace que des techniques classiques, dépendantes de la distribution des points sur des surfaces. En effet, notre algorithme se base sur les symétries partielles dominantes des modèles pour identifier le repère canonique d'alignement. Alors que l'utilisation de la symétrie globale était appliquée avec succès sur les maillages dans les travaux précédents, l'utilisation de la symétrie

locale associée à la projection orthographique est une approche innovante. De plus, notre algorithme d'alignement est appliqué à l'indexation des modèles 3D comme une tâche de prétraitement. En utilisant le descripteur de Fourier normalisé des contours 2D, les résultats obtenus de nos expérimentations montrent une efficacité significative. Nous avons aussi proposé une approche de la compression des modèles BRep-NURBS en utilisant la réflexion dominante.

2 Perspectives

De ce travail, nous proposons quelques pistes de recherche à poursuivre pour la détection de similarités et ses applications.

2.1 Détection de similarités

À partir des limitations analysées dans la section 8, nous voulons développer une autre approche pour résoudre le problème des *loops* circulaires. En fait, Pekerman et al. [Pekerman 2005] proposent une approche pour mesurer les similarités de deux courbes polynomiales paramétrées différemment. Le résultat de cette recherche nous donne un autre outil pour mettre en correspondance deux faces en se basant sur leurs *loops* extérieures. C'est-à-dire, au lieu d'utiliser des sommets, nous considérons les courbes 3D des *arêtes* pour mettre en correspondance des faces.

Les résultats intéressants de Li [Li 2014], sur la détection de symétrie dans les modèles BRep, sont aussi une bonne piste de recherche. La notion de *faces maximales* peut résoudre notre problème où une face est similaire à une autre face virtuelle qui se composent de plusieurs faces. Car les faces maximales regroupent toutes les faces adjacentes en une seule représentation.

Faute de temps, nous n'avons pas réalisé une évaluation approfondie sur la complexité et la robustesse de notre algorithme proposé et une comparaison avec les autres algorithmes dans la littérature.

2.2 Applications basées sur les similarités détectées

Pour l'alignement des modèles BRep-NURBS, nous n'avons pas trouvé une *base de référence* (benchmark) de ces modèles pour comparer la robustesse et l'efficacité de notre algorithme avec les autres. En effet, il y a de nombreux travaux sur l'alignement des maillages mais nous n'avons pas trouvé un travail concernant l'alignement des modèles BRep-NURBS. De plus, notre algorithme de détection de similarités peut identifier les isométries autres que la symétrie telles que les rotations et les translations. Dans un travail, Chaouch et al. [Chaouch 2008] utilise l'invariance translationnelle locale associée avec la réflexion pour aligner ses modèles. Alors, est-ce que les autres isométries peuvent servir à l'alignement de nos modèles? Cela reste une autre piste de recherche.

Les similarités détectées peuvent servir à l'édition de forme, cette application peut fournir aux concepteurs 3D une interface pour générer les nouveaux modèles

à partir des modèles existants. À tout moment, un concepteur souhaite altérer de façon uniforme un certain nombre de régions similaires. Il lui suffit de modifier une seule région, les autres régions similaires seront synchronisées.

De même, les similarités détectées peuvent servir à la compression des modèles, permettant un stockage des modèles réduits. L'utilisation de tout type d'isométries peut améliorer le ratio de compression actuel. Cela peut faciliter le streaming des modèles dans les réseaux aux ressources restreintes.

Bibliographie

- [Arun 1987] K.S. Arun, T.S. Huang et S.D. Blostein. *Least Squares Fitting of two 3D Point Sets*. IEEE transactions on Pattern Analysis and Machine Intelligence, vol. PAMI-9, no. 5, pages 698–700, 1987. (Cité en pages 58 et 59.)
- [Atallah 1985] Mikhail J Atallah. *On symmetry detection*. Computers, IEEE Transactions on, vol. 100, no. 7, pages 663–666, 1985. (Cité en page 44.)
- [Bajaj 1997] Chandrajit Bajaj, H Y Lee, R Merkert et Valerio Pascucci. *NURBS based B-rep models for macromolecules and their properties*. In Proceedings of the fourth ACM symposium on Solid modeling and applications, pages 217–228. ACM, 1997. (Cité en pages 5 et 15.)
- [Berner 2008] Alexander Berner, Martin Bokeloh, Michael Wand, Andreas Schilling et Hans-Peter Seidel. *A Graph-Based Approach to Symmetry Detection*. In Symposium on Volume and Point-Based Graphics, pages 1–8, Los Angeles, CA, 2008. Eurographics Association. (Cité en page 52.)
- [Bokeloh 2009] M Bokeloh, A Berner, M Wand, H P Seidel et A Schilling. *Symmetry Detection Using Feature Lines*. EUROGRAPHICS, vol. 28, no. 2, 2009. (Cité en page 52.)
- [Campello 2009] R.J.G.B. Campello et E R Hruschka. *On comparing two sequences of numbers and its applications to clustering analysis*. Information Sciences, vol. 179, no. 8, pages 1025–1039, 2009. (Cité en page 64.)
- [Catmull 1978] E. Catmull et J. Clark. *Recursively generated B-spline surfaces on arbitrary topological meshes*, 1978. (Cité en page 13.)
- [Caumon 2003] Guillaume Caumon, Charles H Sword Jr. et Jean-Laurent Mallet. *Constrained Modifications of Non-manifold B-reps*. In Proceedings of the Eighth ACM Symposium on Solid Modeling and Applications, SM '03, pages 310–315, New York, NY, USA, 2003. ACM. (Cité en pages 5 et 15.)
- [Cederberg 2001] Judith N Cederberg. *A course in modern geometries*. Springer, 2001. (Cité en pages 6 et 38.)
- [Chaouch 2008] Mohamed Chaouch et Anne Verroust-Blondet. *A Novel Method for Alignment of 3D Models*. Research Report RR-6408, INRIA, 2008. (Cité en pages 45, 85, 86 et 101.)
- [Chu 2006] Chih-Hsing Chu et Yung-Chang Hsu. *Similarity assessment of 3D mechanical components for design reuse*. Robotics and Computer-Integrated Manufacturing, vol. 22, no. 4, pages 332–341, 2006. (Cité en pages 5 et 15.)
- [Comaniciu 1999] Dorin Comaniciu et Peter Meer. *Mean shift analysis and applications*. In Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on, volume 2, pages 1197–1203. IEEE, 1999. (Cité en pages vi, 66, 67 et 68.)

- [Comaniciu 2002] Dorin Comaniciu, Peter Meer et Senior Member. *Mean shift : A robust approach toward feature space analysis*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 24, no. 5, pages 603–619, 2002. (Cité en pages 66 et 67.)
- [Comaniciu 2003] Dorin Comaniciu. *An algorithm for data-driven bandwidth selection*. Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol. 25, no. 2, pages 281–288, 2003. (Cité en page 68.)
- [Cuillière 2011] Jean-Christophe Cuillière, Vincent François, Khaled Souaissa, Abdelmajid Benamara et Hedi BelHadjSalah. *Automatic comparison and remodeling applied to {CAD} model modification*. Computer-Aided Design, vol. 43, no. 12, pages 1545–1560, 2011. (Cité en page 5.)
- [DeRose 1998] Tony DeRose, Michael Kass et Tien Truong. *Subdivision surfaces in character animation*. Proceedings of the 25th annual conference on Computer graphics and interactive techniques SIGGRAPH 98, vol. pages, no. Annual Conference Series, pages 85–94, 1998. (Cité en pages 1 et 13.)
- [Dimas 1999] E Dimas et D Briassoulis. *3D geometric modelling based on NURBS : a review*. Advances in Engineering Software, vol. 30, no. 9-11, pages 741–751, 1999. (Cité en pages 2, 4 et 15.)
- [Ducci 2003] Diego Santa Cruz Ducci. *Compression of 3D Models with NURBS*. PhD thesis, {É}COLE POLYTECHNIQUE F{É}D{É}RALE DE LAUSANNE, 2003. (Cité en page 95.)
- [Dugelay 2008] Jean-Luc Dugelay, Atila Baskurt et Mohamed Daoudi. *3D object processing : compression, indexing and watermarking*. John Wiley & Sons, 2008. (Cité en pages 4 et 9.)
- [Eitz 2012] Mathias Eitz, Ronald Richter, Tamy Boubekeur, Kristian Hildebrand et Marc Alexa. *Sketch-based shape retrieval*, 2012. (Cité en page 92.)
- [El-Mehalawi 2003] Mohamed El-Mehalawi et R Allen Miller. *A database system of mechanical components based on geometric and topological similarity. Part I : representation*. Computer-Aided Design, vol. 35, no. 1, pages 83–94, 2003. (Cité en page 4.)
- [Elkott 2002] Diao F Elkott, Hoda A Elmaraghy et Waguih H Elmaraghy. *Automatic sampling for CMM inspection planning of free-form surfaces*. International Journal of Production Research, vol. 40, no. 11, pages 2653–2676, 2002. (Cité en page 22.)
- [Farin 1992] Gerald Farin. *Courbes et surfaces pour la CGAO - Conception Géométrique Assistée par Ordinateur*. Masson, 1992. (Cité en pages 1, 12 et 60.)
- [Ferguson 2000] Ronald W Ferguson. *Modeling orientation effects in symmetry detection : The role of visual structure*. In Proceedings of the twenty-second annual conference of the cognitive science society, pages 125–130, 2000. (Cité en page 85.)

- [Forsey 1988] David R Forsey et Richard H Bartels. *Hierarchical B-spline refinement*. In ACM SIGGRAPH Computer Graphics, volume 22, pages 205–212. ACM, 1988. (Cité en page 95.)
- [Fresnel 2010] Jean Fresnel. Méthodes modernes en géométrie. Hermann, 2010. (Cité en pages 29 et 38.)
- [Furukawa 2002] Yoshiyuki Furukawa et Hiroshi Masuda. *Compression of NURBS surfaces with error evaluation*. NICOGRAPH International, 2002. (Cité en pages vi et 96.)
- [Goshtasby 2005] A Ardeshir Goshtasby. 2-D and 3-D Image Registration : For Medical, Remote Sensing, and Industrial Applications. Wiley-Interscience, 2005. (Cité en page 57.)
- [Hel-Or 2010] Hagit Hel-Or et Craig S Kaplan. Computational symmetry in computer vision and computer graphics. Now publishers Inc, 2010. (Cité en pages 6, 44 et 85.)
- [Huttenlocher 1993] D P Huttenlocher, G A Klanderman et W J Rucklidge. *Comparing images using the Hausdorff distance*. Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol. 15, no. 9, pages 850–863, Septembre 1993. (Cité en page 76.)
- [Iyer 2005] Natraj Iyer, Subramaniam Jayanti, Kuiyang Lou, Yagnanarayanan Kalyanaraman et Karthik Ramani. *Three-dimensional shape searching : state-of-the-art review and future trends*. Computer-Aided Design, vol. 37, no. 5, pages 509–530, Avril 2005. (Cité en pages 5 et 83.)
- [Jesorsky 2001] Oliver Jesorsky, Klaus J. Kirchberg et Robert W. Frischholz. *Robust Face Detection Using the Hausdorff Distance*. In Josef Bigun et Fabrizio Smeraldi, éditeurs, Audio- and Video-Based Biometric Person Authentication, volume 2091 of *Lecture Notes in Computer Science*, pages 90–95. Springer Berlin Heidelberg, 2001. (Cité en page 76.)
- [Johan 2011] Henry Johan, Bo Li, Yuanmin Wei et Iskandarsyah. *3D model alignment based on minimum projection area*. The Visual Computer, vol. 27, no. 6-8, pages 565–574, 2011. (Cité en page 86.)
- [Johnson 2001] David Lawrence Johnson. Symmetries. Springer, 2001. (Cité en page 38.)
- [Kazhdan 2003] Michael M Kazhdan, Bernard Chazelle, David P Dobkin, Thomas A Funkhouser et Szymon Rusinkiewicz. *A Reflective Symmetry Descriptor for 3D Models*. Algorithmica, vol. 38, no. 1, pages 201–225, 2003. (Cité en pages v, 45 et 46.)
- [Kazinnik 1997] Roman Kazinnik et Gershon Elber. *Orthogonal Decomposition of Non-Uniform B-spline Spaces using Wavelets*. In Computer Graphics forum, volume 16, pages C27—C38. Wiley Online Library, 1997. (Cité en page 95.)
- [Koller 2004] David Koller, Michael Turitzin, Marc Levoy, Marco Tarini, Giuseppe Croccia, Paolo Cignoni et Roberto Scopigno. *Protected Interactive 3D Gra-*

- phics via Remote Rendering*. ACM Trans. Graph., vol. 23, no. 3, pages 695–703, 2004. (Cité en page 10.)
- [Levoy 2000] Marc Levoy, Kari Pulli, Brian Curless, Szymon Rusinkiewicz, David Koller, Lucas Pereira, Matt Ginzton, Sean Anderson, James Davis, Jeremy Ginsberg et Others. *The digital Michelangelo project : 3D scanning of large statues*. In Proceedings of the 27th annual conference on Computer graphics and interactive techniques, pages 131–144. ACM Press/Addison-Wesley Publishing Co., 2000. (Cité en page 95.)
- [Li 2008] Ming Li, Frank C Langbein et Ralph R Martin. *Detecting approximate symmetries of discrete point subsets*. Computer-Aided Design, vol. 40, no. 1, pages 76–93, 2008. (Cité en page 49.)
- [Li 2011] Ke Li. *Shape Analysis of B-Rep CAD Models to Extract Partial and Global Symmetries*. PhD thesis, 2011. (Cité en pages v, 49, 50 et 51.)
- [Li 2014] K. Li, G. Foucault, J.-C. Léon et M. Trlin. *Fast global and partial reflective symmetry analyses using boundary surfaces of mechanical components*. Computer-Aided Design, vol. 53, pages 70–89, Aot 2014. (Cité en page 101.)
- [Lian 2010] Zhouhui Lian, Paul L Rosin et Xianfang Sun. *Rectilinearity of 3D meshes*. International Journal of Computer Vision, vol. 89, no. 2-3, pages 130–151, 2010. (Cité en page 85.)
- [Liu 2010] Y-J Liu, X Luo, A Joneja, C-X Ma, X-L Fu et D-W Song. *User-Adaptive Sketch-Based 3-D CAD Model Retrieval*. Automation Science and Engineering, IEEE Transactions on, vol. 10, no. 3, pages 783–795, 2010. (Cité en page 92.)
- [Loop 1987] C. Loop. *Smooth subdivision surfaces based on triangles*. PhD thesis, University of Utah, 1987. (Cité en page 13.)
- [Lorusso 1997] A. Lorusso, D.W. Eggert et R.B. Fisher. *Estimating 3-D rigid body transformations : a comparison of four major algorithms*, 1997. (Cité en page 58.)
- [Loy 2006] Gareth Loy et Jan-Olof Eklundh. *Detecting Symmetry and Symmetric Constellations of Features*. In ECCV, pages 508–521, 2006. (Cité en pages v, 46 et 47.)
- [Meila 2001] M Meila et J Shi. *A Random Walks View of Spectral Segmentation*. International Workshop on Artificial Intelligence and Statistics, 2001. (Cité en page 69.)
- [Mills 2001] Bruce I Mills, Frank C Langbein, A David Marshall et Ralph R Martin. *Approximate symmetry detection for reverse engineering*. In Proceedings of the sixth ACM symposium on Solid modeling and applications, pages 241–248. ACM, 2001. (Cité en pages 45 et 49.)
- [Minovic 1993] P Minovic, S Ishikawa et K Kato. *Symmetry identification of a 3-D object represented by octree*. Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol. 15, no. 5, pages 507–514, 1993. (Cité en page 85.)

- [Mitra 2006] N J Mitra, L Guibas et M Pauly. *Partial and Approximate Symmetry Detection for 3D Geometry*. ACM Transactions on Graphics (SIGGRAPH), vol. 25, no. 3, pages 560–568, 2006. (Cité en pages [v](#), [40](#), [46](#), [47](#), [48](#), [52](#), [54](#), [62](#), [63](#), [68](#), [73](#) et [78](#).)
- [Mitra 2012] Niloy J Mitra, Mark Pauly, Michael Wand et Duygu Ceylan. *Symmetry in 3D Geometry : Extraction and Applications*. In EUROGRAPHICS State-of-the-art Report, 2012. (Cité en pages [6](#), [44](#), [66](#) et [85](#).)
- [Miyazaki 2006] Daisuke Miyazaki, Mawo Kamakura, Tomoaki Higo, Yasuhide Okamoto, Rei Kawakami, Takaaki Shiratori, Akifumi Ikari, Shintaro Ono, Yoshihiro Sato, Mina Oya et Others. *3D digital archive of the burghers of calais*. In Interactive Technologies and Sociotechnical Systems, pages 399–407. Springer, 2006. (Cité en page [95](#).)
- [Mouysset 2010] Sandrine Mouysset. *Contributions à l'étude de la classification spectrale et applications*. PhD thesis, Institut National Polytechnique de Toulouse, 2010. (Cité en page [71](#).)
- [Mouysset 2011] S Mouysset, J Noailles, D Ruiz et R Guivarch. *On a Strategy for Spectral Clustering with Parallel Computation*. High Performance Computing for Computational Science–VECPAR 2010, pages 408–420, 2011. (Cité en page [73](#).)
- [Napoléon 2010] Thibault Napoléon et Hichem Sahbi. *From 2D silhouettes to 3D object retrieval : contributions and benchmarking*. Journal on Image and Video Processing, vol. 2010, 2010. (Cité en page [86](#).)
- [Ng 2002] A Y Ng, M I Jordan et Y Weiss. *On Spectral Clustering : Analysis and an Algorithm*. Advances in neural information processing systems, vol. 2, pages 849–856, 2002. (Cité en pages [69](#) et [71](#).)
- [Obeidat 2009] SuleimanM. Obeidat et Shivakumar Raman. *An intelligent sampling method for inspecting free-form surfaces*. The International Journal of Advanced Manufacturing Technology, vol. 40, no. 11-12, pages 1125–1136, 2009. (Cité en page [22](#).)
- [Osada 2002] Robert Osada, Thomas Funkhouser, Bernard Chazelle et David Dobkin. *Shape Distributions*. ACM Trans. Graph., vol. 21, no. 4, pages 807–832, 2002. (Non cité.) :2002 :2002 :2002 :2002
- [Ovsjanikov 2008] Maks Ovsjanikov, Jian Sun et Leonidas Guibas. *Global intrinsic symmetries of shapes*. In Computer graphics forum, volume 27, pages 1341–1348. Wiley Online Library, 2008. (Cité en page [44](#).)
- [Papadakis 2007] Panagiotis Papadakis, Ioannis Pratikakis, Stavros Perantonis et Theoharis Theoharis. *Efficient 3D shape matching and retrieval using a concrete radialized spherical projection representation*. Pattern Recognition, vol. 40, no. 9, pages 2437–2452, 2007. (Cité en pages [46](#) et [85](#).)
- [Parzen 1962] E Parzen. *On the estimation of probability density function and mode*. Ann. Math. Statist., vol. 33, pages 1065–1076, 1962. (Cité en page [66](#).)

- [Paul Chew 1989] L Paul Chew. *Constrained delaunay triangulations*. Algorithmica, vol. 4, no. 1-4, pages 97–108, 1989. (Cité en pages 27 et 75.)
- [Pekerman 2005] Diana Pekerman, Joon-Kyung Seong, Gershon Elber et Myung-Soo Kim. *Are Two Curves the Same?* Computer-Aided Design and Applications, vol. 2, no. 1-4, pages 85–94, 2005. (Cité en page 101.)
- [Perlin 1985] Ken Perlin. *An image synthesizer*. ACM Siggraph Computer Graphics, vol. 19, no. 3, pages 287–296, 1985. (Cité en page 13.)
- [Piegl 1987] Leslie Piegl et Wayne Tiller. *Curve and surface constructions using rational B-splines*. Computer-Aided Design, vol. 19, no. 9, pages 485–498, Novembre 1987. (Cité en page 15.)
- [Piegl 1991] L. Piegl. *On NURBS : a survey*. IEEE Computer Graphics and Applications, vol. 11, no. 1, 1991. (Cité en pages 4, 12, 14, 15 et 16.)
- [Podolak 2006] Joshua Podolak, Philip Shilane, Aleksey Golovinskiy, Szymon Rusinkiewicz et Thomas Funkhouser. *A Planar Reflective Symmetry Transform for 3D Shapes*. ACM Transactions on Graphics (Proc. SIGGRAPH), vol. 25, no. 3, Juillet 2006. (Cité en pages 46, 47, 54, 85 et 86.)
- [Rosenblatt 1956] Murray Rosenblatt et Others. *Remarks on some nonparametric estimates of a density function*. The Annals of Mathematical Statistics, vol. 27, no. 3, pages 832–837, 1956. (Cité en page 66.)
- [Schoenberg 1968] I J Schoenberg. *CARDINAL INTERPOLATION AND SPLINE FUNCTIONS*. Rapport technique, DTIC Document, 1968. (Cité en page 1.)
- [Shi 2000] J Shi et J Malik. *Normalized Cuts and Image Segmentation*. Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol. 22, no. 8, pages 888–905, 2000. (Cité en page 69.)
- [Stroud 2011] I Stroud et H Nagy. *Solid Modelling and CAD Systems : How to Survive a CAD System*. SpringerLink : B{ü}cher. Springer, 2011. (Cité en pages 5, 17 et 18.)
- [Sun 1997] Changming Sun et Jamie Sherrah. *3D Symmetry Detection Using The Extended Gaussian Image*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 19, pages 164–168, 1997. (Cité en pages v et 45.)
- [Tate 2003] S.J. Tate et G.E.M. Jared. *Recognising symmetry in solid models*. Computer-Aided Design, vol. 35, no. 7, pages 673–692, Juin 2003. (Cité en pages v, 49, 50 et 51.)
- [Tedjokusumo 2006] Jefry Tedjokusumo et WeeKheng Leow. *Normalization and Alignment of 3D Objects Based on Bilateral Symmetry Planes*. In Tat-Jen Cham, Jianfei Cai, Chitra Dorai, Deepu Rajan, Tat-Seng Chua et Liang-Tien Chia, éditeurs, *Advances in Multimedia Modeling*, volume 4351 of *Lecture Notes in Computer Science*, pages 74–85. Springer Berlin Heidelberg, 2006. (Cité en pages 83 et 85.)
- [Tisseron 1988] Claude Tisseron. *Géométries Affine, Projective et Euclidienne*. Hermann, 1988. (Cité en page 29.)

- [Umeyama 1991] S. Umeyama. *Least-squares estimation of transformation parameters between two point patterns*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 13, no. 4, 1991. (Cité en page 59.)
- [Von Luxburg 2007] U Von Luxburg. *A Tutorial on Spectral Clustering*. Statistics and Computing, vol. 17, no. 4, pages 395–416, 2007. (Cité en pages 69 et 70.)
- [Vranić 2004] Dejan V Vranić. *3D model retrieval*. University of Leipzig, Germany, PhD thesis, 2004. (Cité en page 84.)
- [Weyl 1952] Hermann Weyl. *Symmetry*. Princeton University Press, 1952. (Cité en page 6.)
- [Zabrodsky 1995] H Zabrodsky, S Peleg et D Avnir. *Symmetry as a Continuous Feature*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 17, pages 1154–1166, 1995. (Cité en pages v, 45, 46 et 47.)
- [Zhang 2001] D Zhang et G Lu. *A comparative study on shape retrieval using Fourier descriptors with different shape signatures*. In Proc. of international conference on intelligent multimedia and distance education (ICIMADE01), pages 1–9, 2001. (Cité en pages 92 et 93.)
- [Zhao 2013] Shanghong Zhao, Wei Tsang Ooi, Axel Carlier, Geraldine Morin et Vincent Charvillat. *3D mesh preview streaming*. In Proceedings of the 4th ACM Multimedia Systems Conference, pages 178–189. ACM, 2013. (Cité en page 95.)